Caderno de Física

Matemática I

- Aula 02: Coeficientes de Fourier

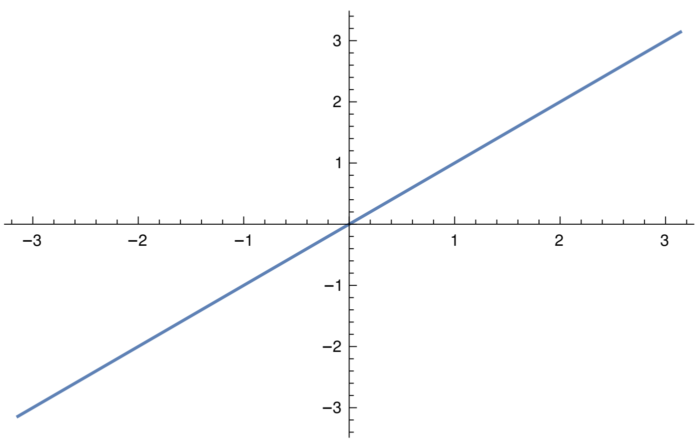
Usamos as funções FourierSinCoefficient[função, x, n] e FourierCosCoefficient[função, x, n] para calcular os coeficientes de fourier, devemos prestar atenção que o Mathematica calcula esses coeficientes com os limites da integral indo de $[0, \pi]$.

In[ ]:=
```
FourierCosCoefficient[função, x , n]
```
coeficiente de Fourier em cosseno
```
FourierSinCoefficient[função, x, n]
```
coeficiente de Fourier em seno
```
Plot[x, {x, -π, π}]
```
gráfico

Out[ ]= 0

Out[ ]= $-\frac{2\left(-1+(-1)^n\right)\text{função}}{n\pi}$

Out[ ]=



Observamos durante as aulas que:

$a_n = 0\ ;\ b_n \neq 0$

In[ ]:=
```
FourierSinCoefficient[x, x, n]
```
coeficiente de Fourier em seno

Out[ ]= $-\frac{2(-1)^n}{n}$

In[ ]:= FourierCosCoefficient[x, x, n]
coeficiente de Fourier em cosseno

$$\frac{2\left(-1+(-1)^n\right)}{n^2 \pi}$$

O resultado acima está errado, o coeficiente foi integrado no período [0,π],sendo que o correto seria [-π,π].No período certo o resultado seria zero.

In[ ]:= $\frac{1}{\pi}$ * Integrate[x * Cos[n x], {x, -π, π}, Assumptions → n ∈ Integers]
integra cosseno premissas números intei

Out[ ]= 0

In[ ]:= $\frac{1}{\pi}$ * Integrate[x * Sin[n x], {x, -π, π}, Assumptions → n ∈ Integers]
integra seno premissas números intei

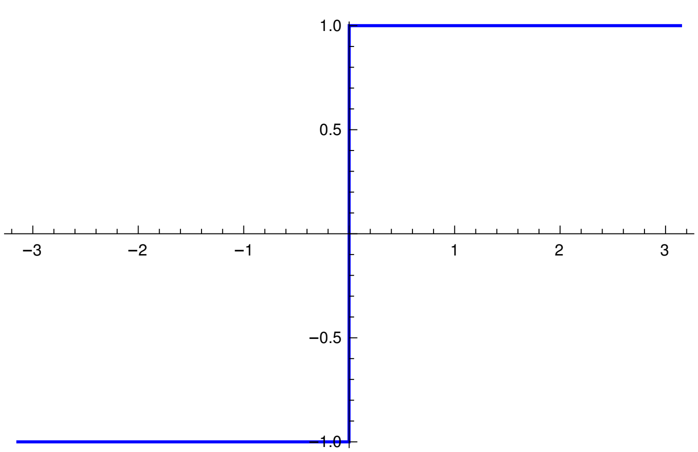
Out[ ]= $$\frac{2\,(-n\,\pi\,\mathrm{Cos}[n\,\pi]+\mathrm{Sin}[n\,\pi])}{n^2\,\pi}$$

In[ ]:= FullSimplify[%, n ∈ Integers]
simplifica completamente números inte

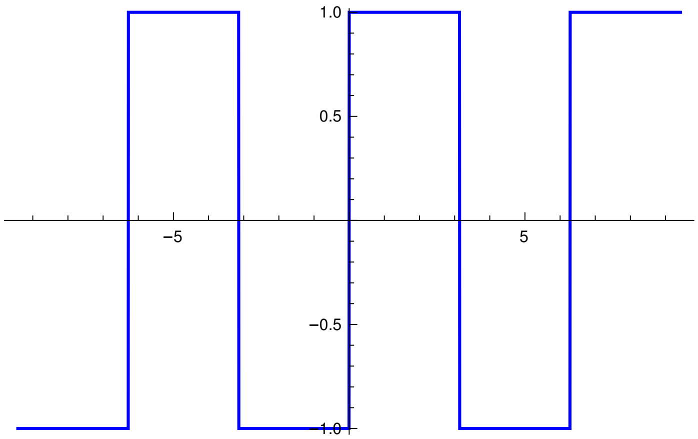
Out[ ]= $$-\frac{2\,(-1)^n}{n}$$

Seja f[x] uma função qualquer. Para deixa-la periódica por partes, com período L, use f[Mod[x,L,-L/2]]. A função Mod calcula x módulo L. O terceiro argumento é o offset, quando ele é -L/2, estamos pegando a função entre [-L/2, L/2]. Se o terceiro argumento fosse 0 então repetiríamos a função dentro do intervalo [0, L].
Abaixo veremos os exemplos estudados em aula:

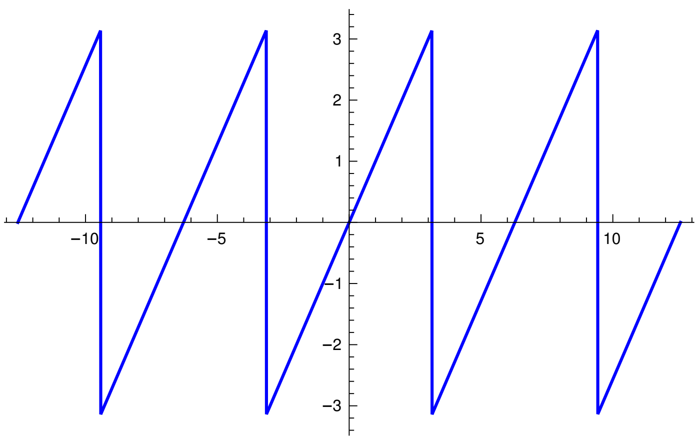
In[ ]:= Plot[Sign[x], {x, -π, π}, Exclusions → None, PlotRange → {-1.03, 1.02}, PlotStyle → Blue]
gráf⋯ função de sinal exclusões nen⋯ intervalo do gráfico estilo do gráfico azul

Out[ ]=


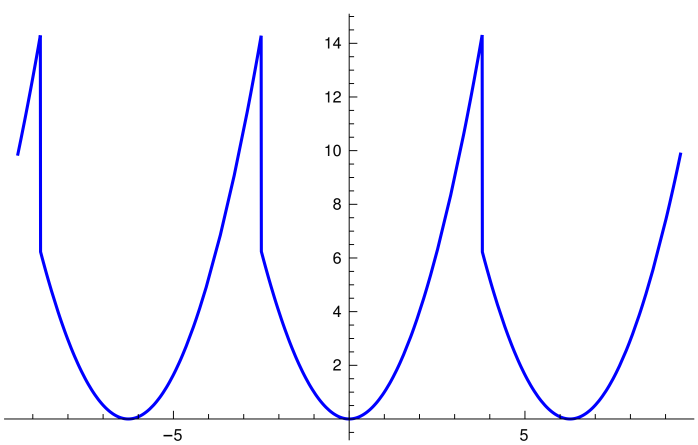
In[ ]:=
```
Plot[Sign[Mod[x, 2 π, -π]], {x, -3 π, 3 π},
  Exclusions → None, PlotRange → {-1.03, 1.02}, PlotStyle → Blue]
```
gráf·· fun··· operação do módulo
exclusões nen··· intervalo do gráfico estilo do gráfico azul

Out[ ]=


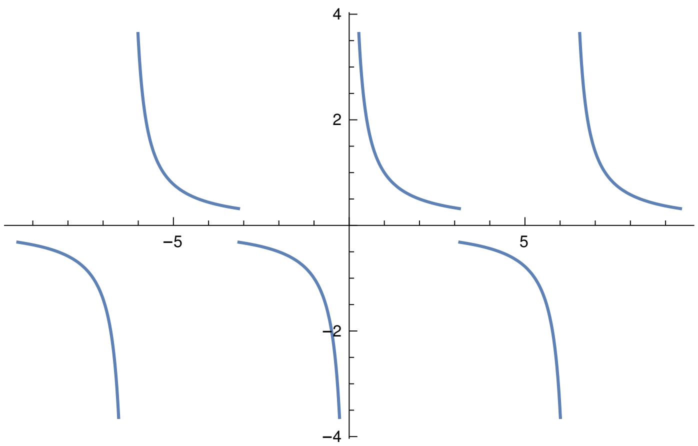
In[ ]:=
```
Plot[Mod[x, 2 π, -π], {x, -4 π, 4 π}, PlotStyle → Blue, Exclusions → None]
```
gráf·· operação do módulo estilo do gráfico azul exclusões nenhun

Out[ ]=


In[ ]:=
```
Plot[Mod[x, 2 π, -2.5]^2, {x, -3 π, 3 π}, PlotStyle → Blue]
```
gráfico estilo do gráfico azul

Out[ ]=


In[ ]:=
```
meuplot = Plot[1/Mod[x, 2 π, -π], {x, -3 π, 3 π}]
```
gráfico

Out[ ]=



```
(* Para salvar um gráfico;
ou clique no gráfico e vá em "Save Selection As"*)
```
salva

In[ ]:=
```
Export["filepath.pdf", meuplot]
```
exporta

Out[ ]= filepath.pdf

In[ ]:=
```
FourierSinCoefficient[Sign[x], x, n]
```
coeficiente de Fourier em seno | função de sinal

Out[ ]= $-\frac{2\,(-1+(-1)^{n})}{n\,\pi}$

In[ ]:=
```
(* n indo de 1 até 20 em passos de 2 *)
Sum[4/(n π) Sin[n x], {n, 1, 20, 2}]
```
soma | seno

Out[ ]= $\frac{4\,\mathrm{Sin}[x]}{\pi}+\frac{4\,\mathrm{Sin}[3\,x]}{3\,\pi}+\frac{4\,\mathrm{Sin}[5\,x]}{5\,\pi}+\frac{4\,\mathrm{Sin}[7\,x]}{7\,\pi}+\frac{4\,\mathrm{Sin}[9\,x]}{9\,\pi}+\frac{4\,\mathrm{Sin}[11\,x]}{11\,\pi}+\frac{4\,\mathrm{Sin}[13\,x]}{13\,\pi}+\frac{4\,\mathrm{Sin}[15\,x]}{15\,\pi}+\frac{4\,\mathrm{Sin}[17\,x]}{17\,\pi}+\frac{4\,\mathrm{Sin}[19\,x]}{19\,\pi}$

In[ ]:=
```
Sum[4/(n π) Sin[n x], {n, 1, 20, 2}]
soma seno

Sum[1/(n π) [Sin[n * v] - Sin[u * n]] * Cos[n * x], {n, 1, 4}] -
soma seno seno cosseno

Sum[1/(n π) [Cos[n * v] - Cos[u * n]] * Sin[n * x], {n, 1, 4}]
soma cosseno cosseno seno
```

Out[ ]=
$$\frac{4\,\text{Sin}[x]}{\pi} + \frac{4\,\text{Sin}[3\,x]}{3\,\pi} + \frac{4\,\text{Sin}[5\,x]}{5\,\pi} + \frac{4\,\text{Sin}[7\,x]}{7\,\pi} + \frac{4\,\text{Sin}[9\,x]}{9\,\pi} + \frac{4\,\text{Sin}[11\,x]}{11\,\pi} + \frac{4\,\text{Sin}[13\,x]}{13\,\pi} + \frac{4\,\text{Sin}[15\,x]}{15\,\pi} + \frac{4\,\text{Sin}[17\,x]}{17\,\pi} + \frac{4\,\text{Sin}[19\,x]}{19\,\pi}$$

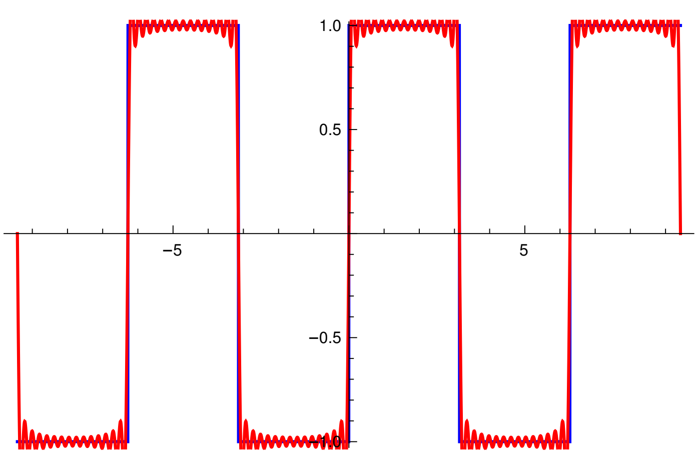
Out[ ]=
$$\text{Cos}[4\,x]\,\frac{1}{4\,\pi}[2\,\text{Sin}[4]] + \text{Cos}[3\,x]\,\frac{1}{3\,\pi}[2\,\text{Sin}[3]] + \text{Cos}[2\,x]\,\frac{1}{2\,\pi}[2\,\text{Sin}[2]] + \text{Cos}[x]\,\frac{1}{\pi}[2\,\text{Sin}[1]] - \frac{1}{\pi}[0]\,\text{Sin}[x] - \frac{1}{2\,\pi}[0]\,\text{Sin}[2\,x] - \frac{1}{3\,\pi}[0]\,\text{Sin}[3\,x] - \frac{1}{4\,\pi}[0]\,\text{Sin}[4\,x]$$

In[ ]:=
```
Plot[{
gráfico

  Sign[Mod[x, 2 π, -π]],
  fun··· operação do módulo

  Sum[4/(n π) Sin[n x], {n, 1, 30, 2}]
  soma seno

}, {x, -3 π, 3 π}, Exclusions → None, PlotRange → {-1.03, 1.02}, PlotStyle → {Blue, Red}]
                   exclusões        nen···    intervalo do gráfico          estilo do gráfico  azul  vermel
```

Out[ ]=


In[ ]:=
```
? Mod
```

Symbol

Mod[m, n] gives the remainder on division of m by n.

Mod[m, n, d] uses an offset d.

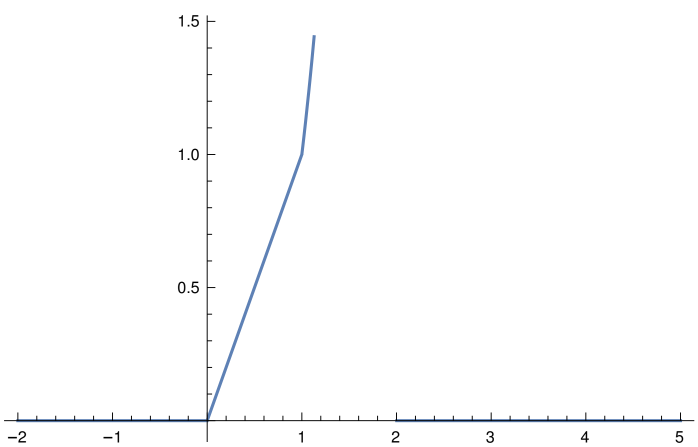
Aula 02 : Exemplos de Serie de Fourier

Alguns exemplos de funções definidas por partes:

In[ ]:=
```
f = Piecewise[{
    função por partes
   {x, 0 < x < 1},
   {x^3, 1 < x < 2}
  }]
```

Out[ ]=
$$\begin{cases} x & 0 < x < 1 \\ x^3 & 1 < x < 2 \\ 0 & \text{True} \end{cases}$$

In[ ]:=
```
Plot[f, {x, -2, 5}]
gráfico
```

Out[ ]=


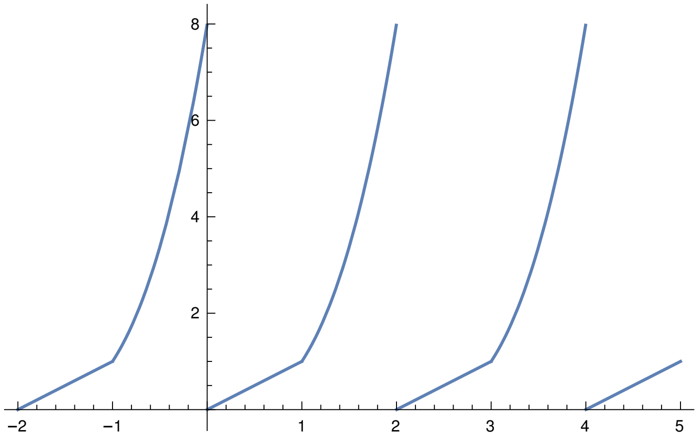
In[ ]:=
```
Clear[f];
apaga
```

In[ ]:=
```
f[x_] := Piecewise[{
          função por partes
   {x, 0 < x < 1},
   {x^3, 1 < x < 2}
  }]
```

Mod[x,L]:   x periódica entre [0,L]

Mod[x,L,d]: x periódica [-d,L-d]

In[ ]:=
```
Plot[f[Mod[x, 2]], {x, -2, 5}]
```
gráfico   operação do módulo

Out[ ]=


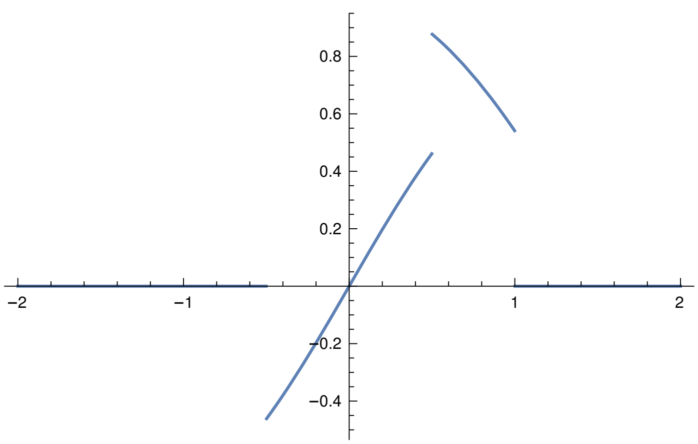
In[ ]:=
```
Clear[f];
```
apaga

```
f[x_] := Piecewise[{
    {Tanh[x], -1/2 < x < 1/2},
    {Cos[x], 1/2 < x < 1}
  }]
```
função por partes

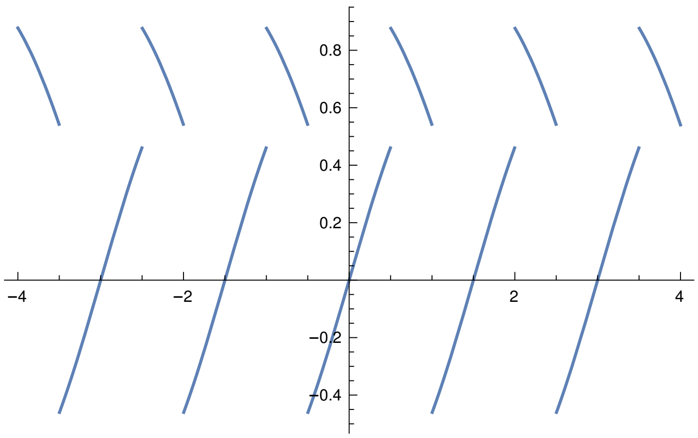
tangente hiperbólica

cosseno

In[ ]:=
```
Plot[f[x], {x, -2, 2}]
```
gráfico

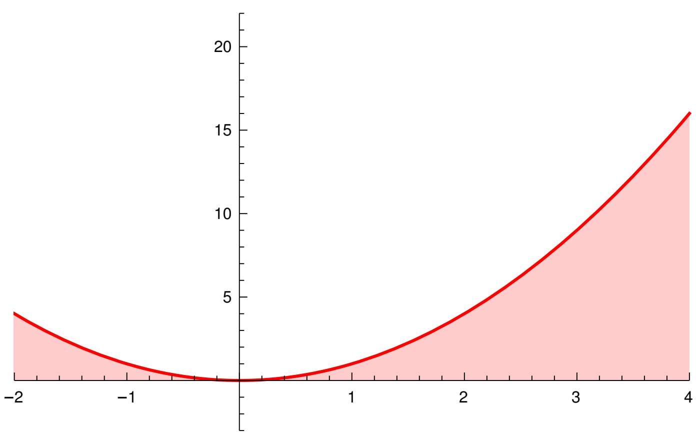
Out[ ]=


In[ ]:=
```
Plot[f[Mod[x, 3 / 2, -1 / 2]], {x, -4, 4}]
```
gráfico operação do módulo

Out[ ]=


Customização de gráficos

In[ ]:=
```
Plot[x^2, {x, -4, 4},
 PlotRange → {{-2, 4}, {-3, 22}},
 PlotStyle → Red,
 Filling → Axis
]
```
gráfico
intervalo do gráfico
estilo do gráfico vermelho
coloração eixo

Out[ ]=


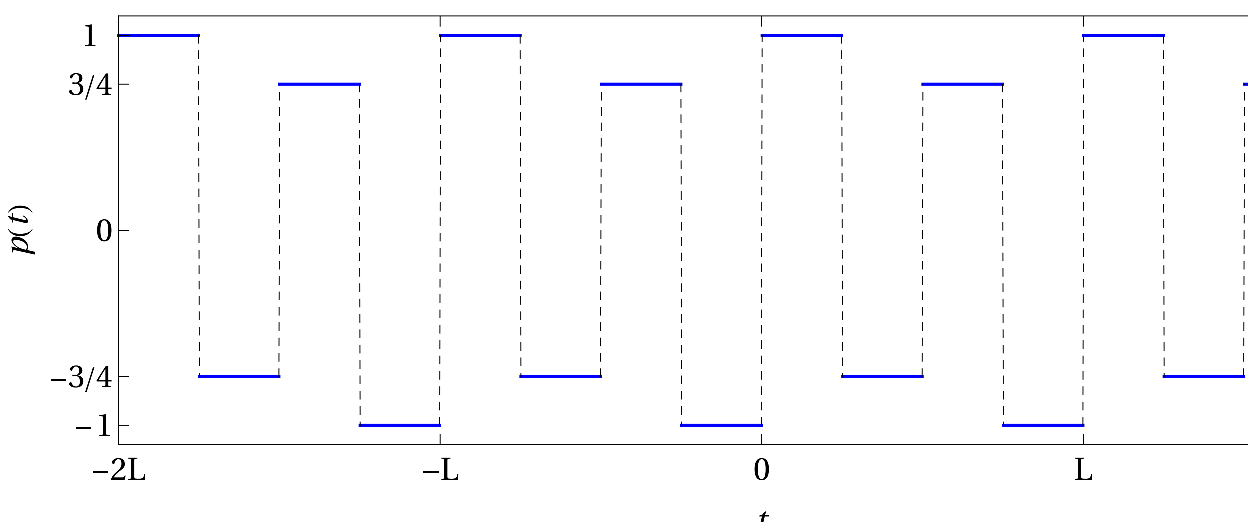
- Aula 07: Fourier and Sound

In[ ]:=
```
SetDirectory[NotebookDirectory[]];
define diretório  diretório do notebook
SetOptions[Plot, ImagePadding → {{60, 20}, {40, 10}}, ExclusionsStyle → Dashed,
define opções  gráfico  preenchimento de imagem  estilo de exclusões  tracejado
  AspectRatio → 1/3, ImageSize → 750, PlotStyle → Blue, Frame → True];
  quociente de aspecto  tamanho da imagem  estilo do gráfico  azul  quadro  verdadeiro
```

Application of Fourier series to sound

Sound propagates as pressure waves in air.
When this wave reaches our ears, it causes tiny bones in the inner ear to wiggle.
These, in turn, agitate a fluid which is then detected by hair cells that convert this movement into electrical pulses.

Hearing is therefore associated with pressure waves.
Let $p(t)$ describe the variations of the pressure, around the ambient value, that arrive in our ears.
- The *tone* of the sound will be associated to the frequency of $p(t)$.
- The *intensity* of the sound will be associated with the absolute value $|\, p(t)\, |^2$

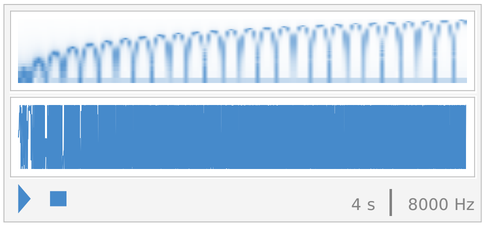
Out[ ]=


Here is a psychedelic sound:

In[ ]:= `Play[Sin[300 t Sin[20 t]], {t, 0, 4}]`

rep ··· seno seno

Out[ ]=



Pure musical notes correspond to well defined sinusoidal functions

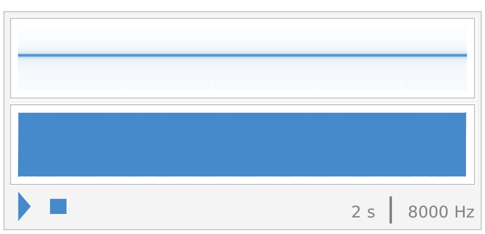
	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

Note A in the middle of the piano, for instance, is $A_4 = 440$ Hz

In[ ]:= `Play[Sin[440 × 2 π t], {t, 0, 2}]`

rep ··· seno

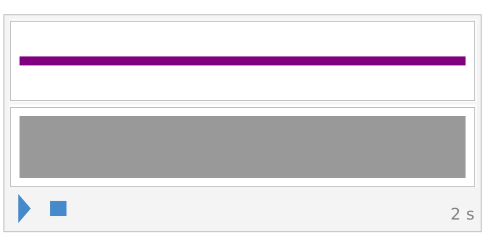
Out[ ]=



In[ ]:= `Sound[SoundNote["A", 2, "Piano"]]`

som nota de som

Out[ ]=



When talking about sound, we generally use frequency ν.

The period L appearing in our Fourier analysis is simply

$L = 1/\nu$

so the Fourier series becomes

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(2\pi n \nu t) + b_n \sin(2\pi n \nu t)]$$

Combining different harmonics

Here I am using a pure sine series (only $b_n \neq 0$), with different combinations of Fourier coefficients.

(shift+enter on the code below to run; the “Manipulate” function is very heavy, so it is good practice to delete the output whenever you are not using it).

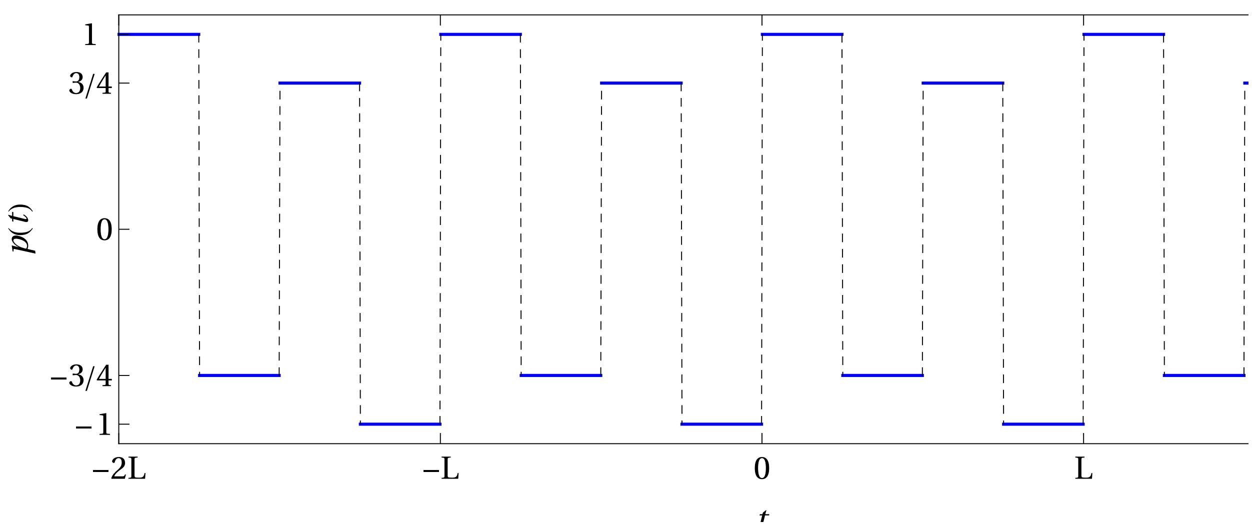
In[ ]:=
```
ν0 = 440;
Manipulate[
 Play[b1 Sin[2 π ν0 t] + b2 Sin[2 (2 π ν0 t)] +
   b3 Sin[3 (2 π ν0 t)] + b4 Sin[4 (2 π ν0 t)] + b5 Sin[5 (2 π ν0 t)], {t, 0, 2}],
 {{b1, 1, "b1"}, 0, 5, 1}, {{b2, 0, "b2"}, 0, 5, 1}, {{b3, 0, "b3"}, 0, 5, 1},
 {{b4, 0, "b4"}, 0, 5, 1}, {{b5, 0, "b5"}, 0, 5, 1}, ControlType → Setter,
 LabelStyle → Directive[22, FontFamily → "Times"]]
```

manipula; reprodu··; seno; seno; seno; seno; seno; tipo de controle; seletor; diretiva; família da fonte; multiplicação

Relative contributions from different harmonics

A sound of frequency ν, will in general contain contributions from different harmonics.
Here is an example of a sound at frequency C_5:

```
In[ ]:= Clear[f, L, v];
        apaga
v = 524; (* C5 *)
L = 1 / v;
h = 3 / 4;
f[x_] := Piecewise[{
         função por partes
   {1, 0 < x < L/4},
   {-h, L/4 < x < 2 L/4},
   {h, 2 L/4 < x < 3 L/4},
   {-1, 3 L/4 < x < L}}]
Plot[f[Mod[x, L]], {x, -2 L, 2 L}, PlotRange → {-1.1, 1.1},
gráfico operação do módulo     intervalo do gráfico
 FrameTicks → {{{-1, {-h, "-3/4"}, 0, {h, "3/4"}, {1, "1  "}}, Automatic},
 marcadores de quadro                                           automático
   {N@{{-2 L, "-2L"}, {-L, "-L"}, {0, "0"}, {L, "L"}, {2 L, "2L"}}, Automatic}},
    valor numérico                                                automático
 FrameLabel → {Style["t", Italic], Style["p(t)", Italic]}
 legenda do quadro estilo      itálico  estilo        itálico
]
```

Out[ ]=


This is not a pure sinusoidal function, however.
It will contain contributions from multiple harmonics.

This is what Fourier series is all about.

Since the pulse is odd, this can be decomposed as a sine series:

$p(t) = \sum_{n=1}^{\infty} b_n \sin(2\,\pi\, n\, \nu\, t)$

The Fourier coefficients are

In[ ]:=
$$\text{coefs} = \text{Table}\left[\frac{2}{L}\,\text{Integrate}\left[f[x]\,\text{Sin}\left[\frac{2\,\pi\,n\,x}{L}\right],\{x,0,L\}\right],\{n,1,10\}\right]$$
tabela integra seno

Out[ ]=
$$\left\{\frac{1}{2\,\pi},\frac{7}{2\,\pi},\frac{1}{6\,\pi},0,\frac{1}{10\,\pi},\frac{7}{6\,\pi},\frac{1}{14\,\pi},0,\frac{1}{18\,\pi},\frac{7}{10\,\pi}\right\}$$

These factors of π make them a bit ugly.
But their absolute value in this case does not matter, since that is related to the overall intensity of the sound.
All that matters are their relative intensities: $|\,b_n\,|^2 \,/\, |\,b_1\,|^2$

In[ ]:=
$$\frac{\text{coefs}^2}{\text{First}[\text{coefs}]^2}$$

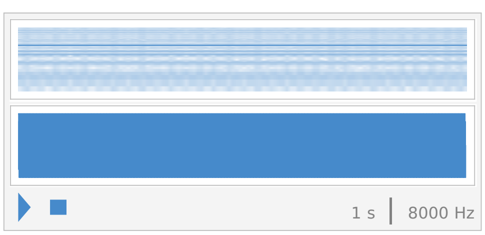
Out[ ]=
$$\left\{1,49,\frac{1}{9},0,\frac{1}{25},\frac{49}{9},\frac{1}{49},0,\frac{1}{81},\frac{49}{25}\right\}$$

Média do pulso sonoro ao longo de um período: identidade de Parseval,

$$\frac{1}{L}\int_0^L |\,p(t)\,|^2\,dt = \sum_{n=1}^{\infty} |\,b_n\,|^2$$

We see that even though the sound has frequency ν, the 2nd harmonic is actually much more important.
It’s contribution is 49 times higher than the principal harmonic!
Other harmonics contribute significantly as well, like the 6th harmonic.

In[ ]:=
```
Play[f[Mod[x, L]], {x, 0, 1}]
```
repro⋯ operação do módulo

Out[ ]=


In[ ]:=
```
(* shift+enter to run this Manipulate *)
                                        manipula
Manipulate[Play[Sin[n 2 π ν t], {t, 0, 1}],
manipula   rep⋯ seno
 {{n, 1}, 1, 10, 1}, ControlType → Setter, LabelStyle → Directive[20, FontFamily → "Times"]]
                     tipo de controle  seletor  estilo de etiqueta  diretiva  família da fonte  multiplicaçã
```

Playing with sounds

Here I define a generalization of the step function used above, so that we can play with the relative Fourier coefficients.

In[ ]:=
```
Clear[f, L, ν, h];
apaga
f[x_, L_, h_] := Piecewise[{
                 função por partes
```

$$\left\{1,\ 0 \le x < \frac{L}{4}\right\},$$

$$\left\{-h,\ \frac{L}{4} < x < \frac{2\,L}{4}\right\},$$

$$\left\{h,\ \frac{2\,L}{4} < x < \frac{3\,L}{4}\right\},$$

$$\left\{-1,\ \frac{3\,L}{4} < x \le L\right\}\}];$$

In[ ]:=
```
bn[n_, v_, h_] := (4 (-1 + h - 2 Cos[n π/2]) Cos[n π] Sin[n π/4]^2)/(n π);

lim = 2/525;
Manipulate[
 L = 1/v;
 coefs = Table[bn[n, v, h], {n, 1, 10}];
 tab = N[coefs^2/First[coefs]^2, 3];

 ttab = TableForm[
   Table[Style[TraditionalForm[t], FractionBoxOptions → {Beveled → True}], {t, tab}],
   TableHeadings → {Table[b_i, {i, 1, 10}], None}, TableSpacing → {3, 10}];

 Grid[{{Plot[f[Mod[x, L], L, h], {x, -lim, lim}, PlotRange → {-1.1, 1.1},
      FrameTicks → {{{-1, -h, 0, h, {1, "1  "}}, Automatic},
        {N@{{-2 L, "-2L"}, {-L, "-L"}, {0, "0"}, {L, "L"}, {2 L, "2L"}}, Automatic}},
      FrameLabel → {Style["t", Italic], Style["p(t)", Italic]}
     ], ttab
   }}]

 , {{v, 525}, 200, 800, 1}, {{h, 3/4}, 0, 1, 1/100}]
```

Just so you know, I computed the coefficients b_n above using the following code:

In[ ]:=
```
2/L Integrate[f[x, L, h] Sin[2 π n x/L], {x, 0, L}, Assumptions → L > 0] /. L → 1/v // Simplify
```

Out[ ]=
$$\frac{4\left(-1+h-2\cos\left[\frac{n\pi}{2}\right]\right)\cos[n\pi]\sin\left[\frac{n\pi}{4}\right]^2}{n\pi}$$

Template for making pretty density plots

This notebook contains a basic snippet of code for plotting pretty ContourPlots using Mathematica. You can use this to check the behavior of the solution of PDEs.

I illustrate the code with the example done in the notes, corresponding to the 1D heat equation with (L = 1)

$u_0(x) = x\left(x^2 - 3x + 2\right)$

The general solution is

u(x,t) = $\sum_{n=1}^{\infty} b_n e^{-i\alpha k_n^2 t} \sin(k_n x)$

where

$b_n = \frac{2}{L}\int_0^L u_0(x)\sin(k_n x)\,dx$

For our example we get (L = 1)

In[ ]:=
```
bn = FullSimplify[2 Integrate[x (x^2 - 3 x + 2) Sin[n π x], {x, 0, 1}], n ∈ Integers]
```
simplifica comple··· integra seno números intei

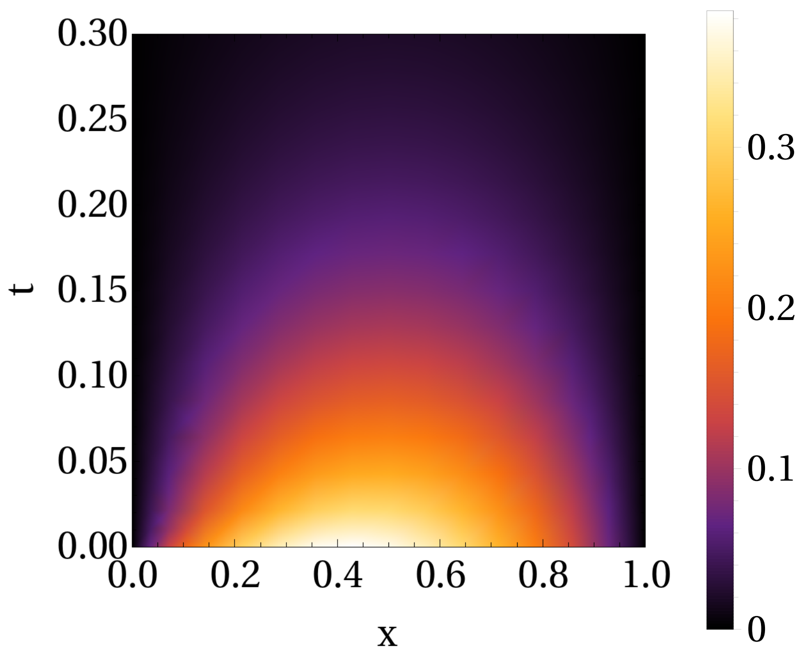
Out[ ]= $\frac{12}{n^3 \pi^3}$

We now use this to define our solution. We take the sum up to a certain large value nmax.
We then plot the result. Everything else are just cosmetic options for the DensityPlot function, which you can play with if you want (α = 1):

In[ ]:=
```
nmax = 300;
sol = Sum[bn Exp[-(n π)^2 t] Sin[n π x], {n, 1, nmax}];
```
soma exponencial seno

In[ ]:=
```
plot = DensityPlot[sol, {x, 0, 1}, {t, 0, 0.3},
  ImageSize → 350,
  FrameLabel → {"x", "t"},
  ColorFunction → "SunsetColors",
  PlotLegends → Automatic,
  PlotRangePadding → None,
  PlotRange → All,
  LabelStyle → {FontFamily → "Times", 20, Black}
 ]
```
gráfico de densidade; tamanho da imagem; legenda do quadro; função de cores; legenda do gráfico; automático; preenchimento de interva··; nenhum; intervalo do gr··; tudo; estilo de etiqueta; família da fonte; multiplicação; preto

Out[ ]=



You can also export them as follows

```
(* Makes the default directory the same as the notebook you are working *)
SetDirectory[NotebookDirectory[]]
```
define diretório   diretório do notebook

```
(* Here "plot" is the name I gave to the plot above. *)
```
aqui   unidade imaginária

```
Export["MyFig.png", plot, ImageResolution → 300];
```
exporta   resolução da imagem

I also recommend that, in the case of Density Plots, you use png instead of PDF.
PDFs tend to get very heavy.

Outros exemplos de condições iniciais sugeridos em aula

Exemplo do Marcio

In[ ]:=
```
bn = FullSimplify[
  2 Integrate[HeavisideTheta[x - 0.4] × HeavisideTheta[0.6 - x] Sin[n π x], {x, 0, 1}],
  n ∈ Integers]
```
simplifica completamente

integra   função teta de Heaviside   função teta de Heaviside   seno

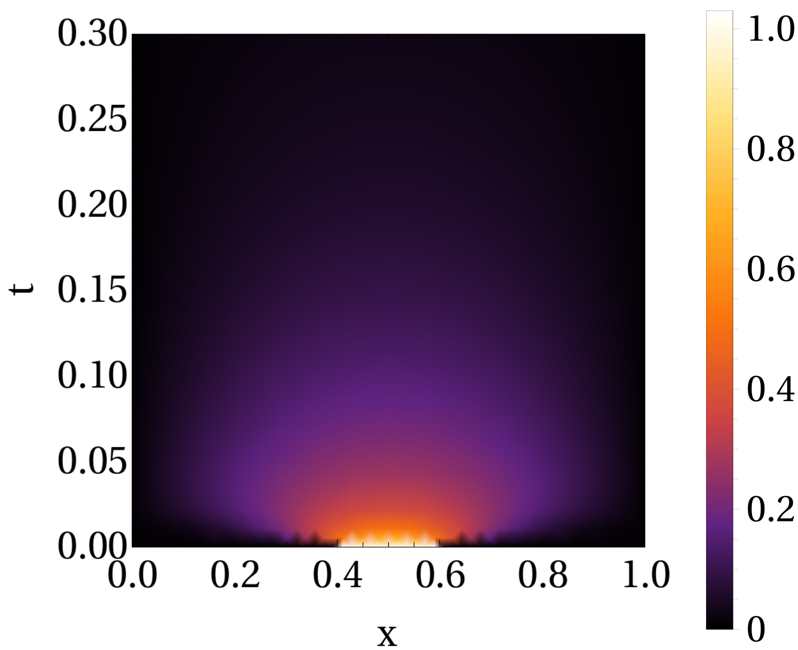
números inteiros

Out[ ]=
$$\frac{0.63662\ \mathrm{Cos}[1.25664\ n] - 0.63662\ \mathrm{Cos}[1.88496\ n]}{n}$$

In[ ]:=
```
nmax = 300;
sol = Sum[bn Exp[-(n π)^2 t] Sin[n π x], {n, 1, nmax}];
```
soma   exponencial   seno

In[ ]:=
```
plot = DensityPlot[sol, {x, 0, 1}, {t, 0, 0.3},
       gráfico de densidade
  ImageSize → 350,
  tamanho da imagem
  FrameLabel → {"x", "t"},
  legenda do quadro
  ColorFunction → "SunsetColors",
  função de cores
  PlotLegends → Automatic,
  legenda do gráfico  automático
  PlotRangePadding → None,
  preenchimento de interva··  nenhum
  PlotRange → All,
  intervalo do gr··  tudo
  LabelStyle → {FontFamily → "Times", 20, Black}
  estilo de etiqueta  família da fonte  multiplicação  preto
 ]
```

Out[ ]=



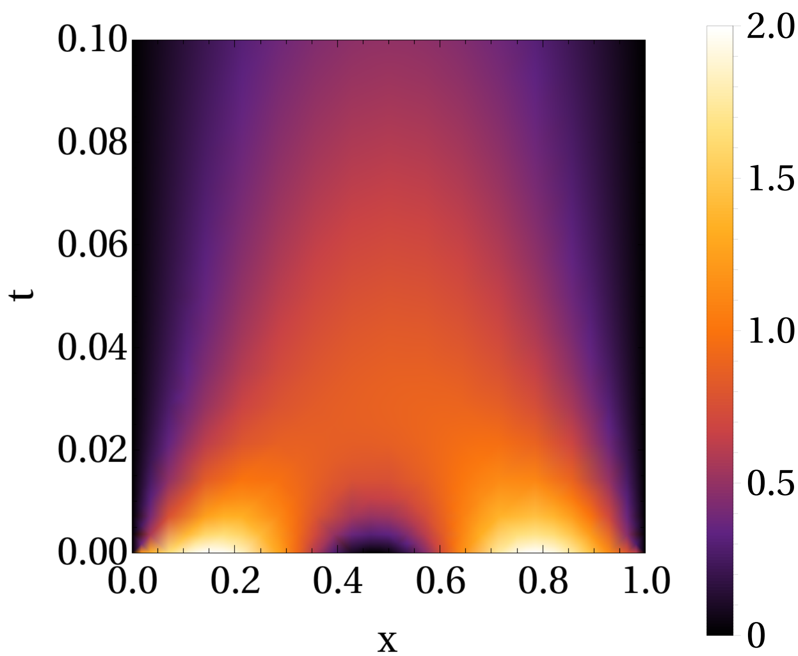
Exemplo do Thiago

In[ ]:=
```
bn = FullSimplify[2 Integrate[(Sin[10 x] + 1) Sin[n π x], {x, 0, 1}], n ∈ Integers]
     simplifica comple···  integra  seno  seno  números inte
```

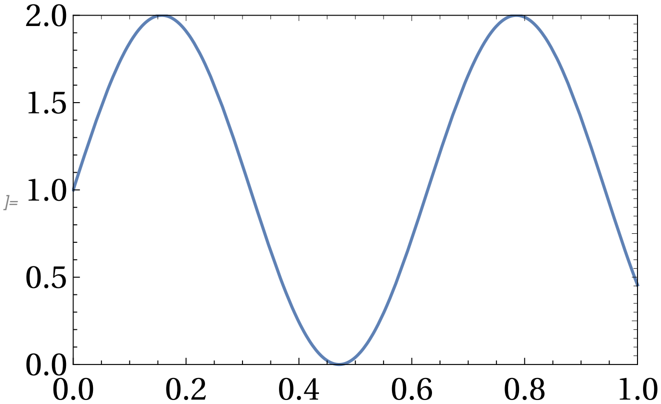
Out[ ]= $$2\left(\frac{1+(-1)^{1+n}}{n\,\pi}+\frac{(-1)^{1+n}\,n\,\pi\,\mathrm{Sin}[10]}{-100+n^2\,\pi^2}\right)$$

In[ ]:=
```
nmax = 300;
sol = Sum[bn Exp[-(n π)^2 t] Sin[n π x], {n, 1, nmax}];
      soma  exponencial      seno
```

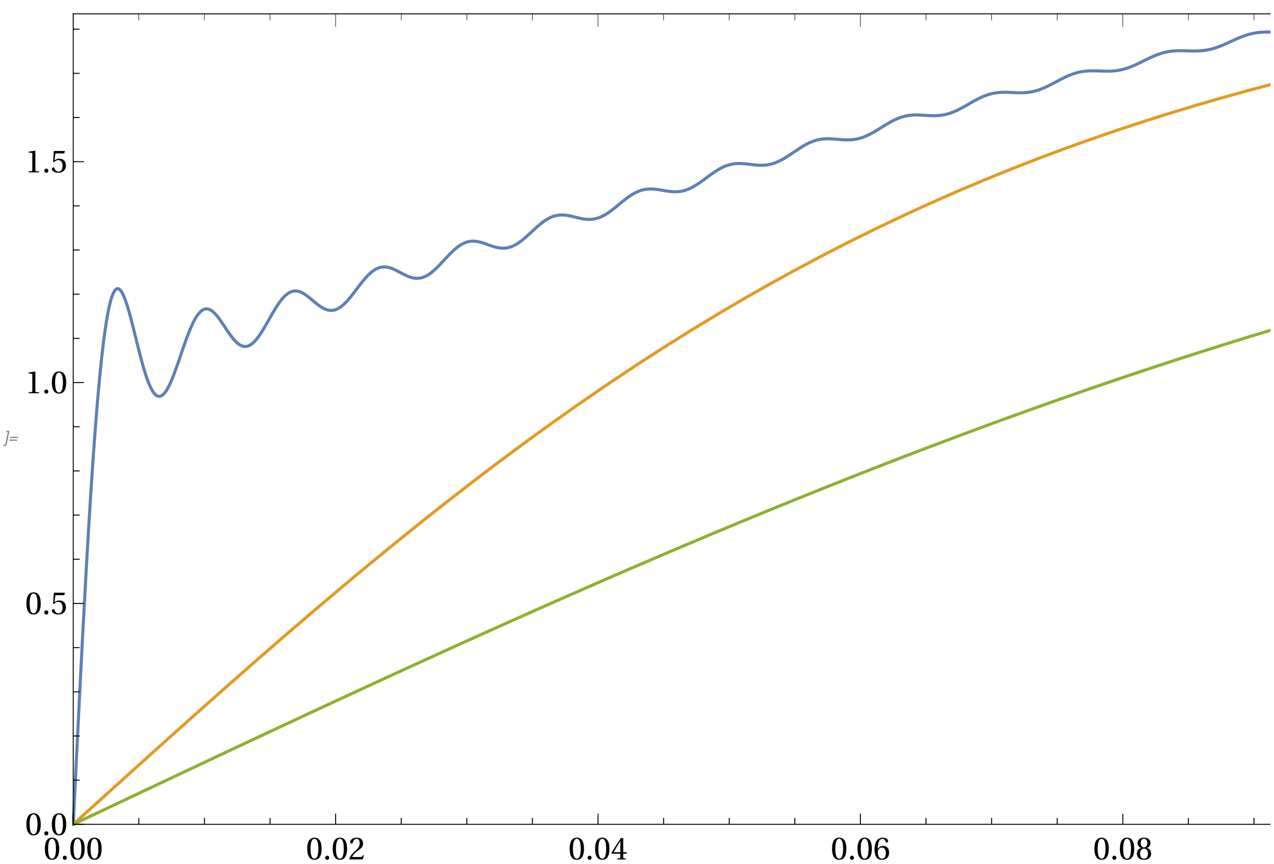
In[ ]:=
```
plot = DensityPlot[sol, {x, 0, 1}, {t, 0, 0.1},
       gráfico de densidade
  ImageSize → 350,
  tamanho da imagem
  FrameLabel → {"x", "t"},
  legenda do quadro
  ColorFunction → "SunsetColors",
  função de cores
  PlotLegends → Automatic,
  legenda do gráfico  automático
  PlotRangePadding → None,
  preenchimento de interva··  nenhum
  PlotRange → All,
  intervalo do gr··  tudo
  LabelStyle → {FontFamily → "Times", 20, Black}
  estilo de etiqueta  família da fonte  multiplicação  preto
 ]
```

Out[ ]=


In[ ]:=
```
Plot[Sin[10 x] + 1, {x, 0, 1}]
```
gráf·· seno

Out[ ]=


In[ ]:=
```
Quiet@Plot[Evaluate@Table[sol, {t, {0, 0.001, 0.005}}], {x, 0, 0.1}]
```
silenci··· gráf·· calcula tabela

Out[ ]=


Exemplo do Luca

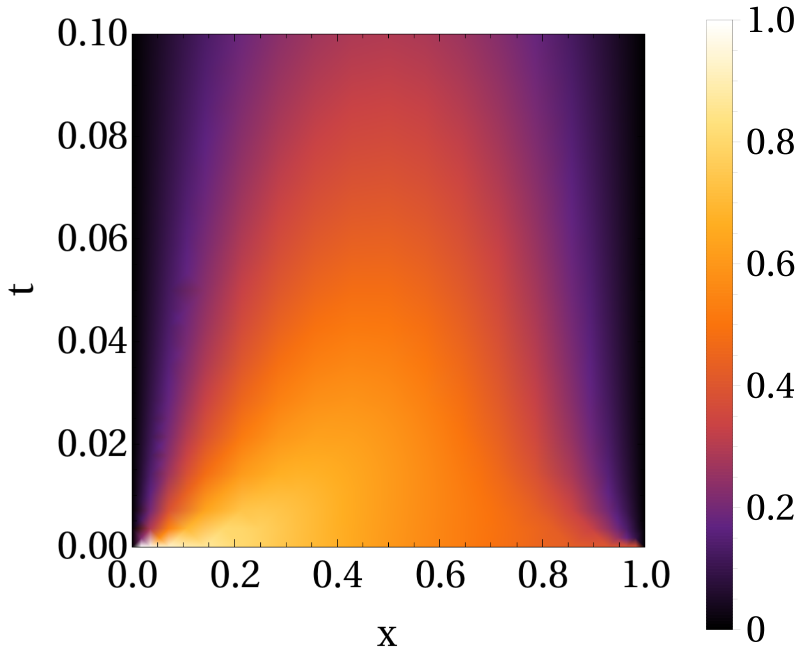
In[ ]:=
```
bn = FullSimplify[2 Integrate[Exp[-x] Sin[n π x], {x, 0, 1}], n ∈ Integers]
```
simplifica comple··· integra expon··· seno números intei

Out[ ]=
$$\frac{2\left((-1)^{1+n}+e\right) n\pi}{e+e\, n^2\pi^2}$$

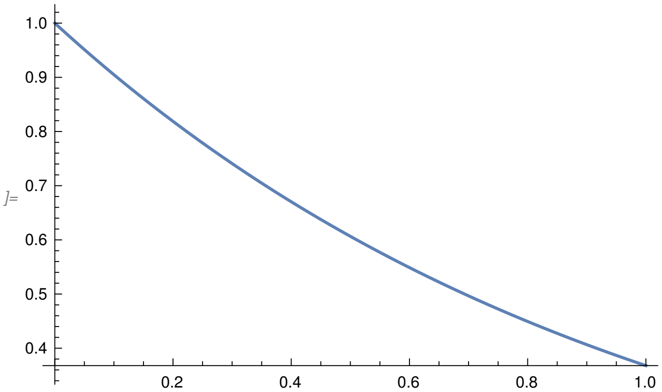
In[ ]:=
```
nmax = 300;
sol = Sum[bn Exp[-(n π)^2 t] Sin[n π x], {n, 1, nmax}];
```
soma exponencial seno

In[ ]:=
```
plot = DensityPlot[sol, {x, 0, 1}, {t, 0, 0.1},
       gráfico de densidade
  ImageSize → 350,
  tamanho da imagem
  FrameLabel → {"x", "t"},
  legenda do quadro
  ColorFunction → "SunsetColors",
  função de cores
  PlotLegends → Automatic,
  legenda do gráfico  automático
  PlotRangePadding → None,
  preenchimento de interva··  nenhum
  PlotRange → All,
  intervalo do gr··  tudo
  LabelStyle → {FontFamily → "Times", 20, Black}
  estilo de etiqueta  família da fonte  multiplicação  preto
 ]
```

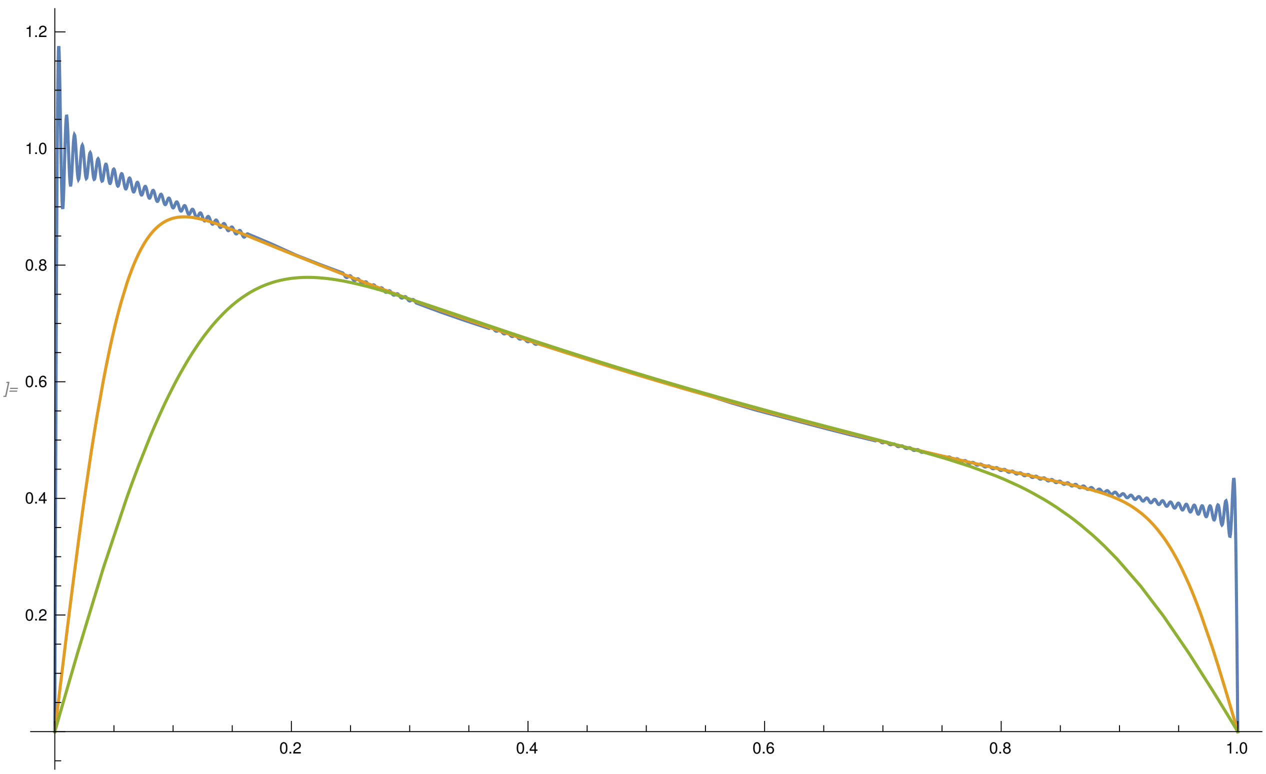
Out[ ]=



In[ ]:= Plot[Exp[-x], {x, 0, 1}]

Out[ ]=


In[ ]:= Quiet@Plot[Evaluate@Table[sol, {t, {0, 0.001, 0.005}}], {x, 0, 1}]

Out[ ]=


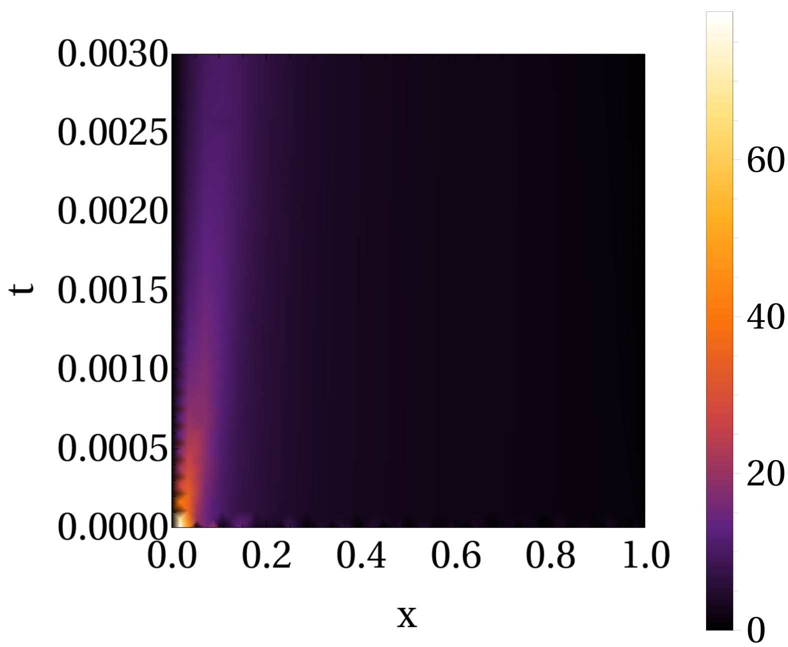
Exemplo do Gabriel

In[ ]:= bn = FullSimplify[2 Integrate[$\frac{1}{x}$ Sin[n π x], {x, 0, 1}], n ∈ Integers]

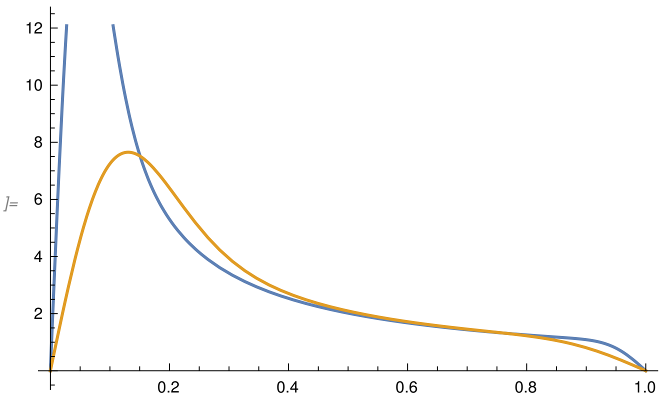
Out[ ]= 2 SinIntegral[n π]

In[ ]:=
```
nmax = 300;
sol = Sum[bn Exp[-(n π)^2 t] Sin[n π x], {n, 1, nmax}];
```
soma exponencial seno

In[ ]:=
```
plot = DensityPlot[sol, {x, 0, 1}, {t, 0, 0.003},
  ImageSize → 350,
  FrameLabel → {"x", "t"},
  ColorFunction → "SunsetColors",
  PlotLegends → Automatic,
  PlotRangePadding → None,
  PlotRange → All,
  LabelStyle → {FontFamily → "Times", 20, Black}
 ]
```
gráfico de densidade; tamanho da imagem; legenda do quadro; função de cores; legenda do gráfico, automático; preenchimento de interva··, nenhum; intervalo do gr··, tudo; estilo de etiqueta, família da fonte, multiplicação, preto

Out[ ]=


In[ ]:=
```
Quiet@Plot[Evaluate@Table[sol, {t, {0.001, 0.005}}], {x, 0, 1}]
```
silenci··· gráf·· calcula tabela

Out[ ]=


In[ ]:=
```
Integrate[α^2 * Exp[-b * x^2], {x, -∞, ∞}]
```
integra exponencial

Out[ ]= ConditionalExpression$\left[\frac{\sqrt{\pi}\ \alpha^2}{\sqrt{b}}, \mathrm{Re}[b] > 0\right]$

Produces animated GIFs for the solution of the 1D wave equation.

Assumes L = c = 1.

In[ ]:=
```
SetDirectory[NotebookDirectory[]];
```
define diretório diretório do notebook

```
anCoefs[f_, nmax_ : 100] := Table[2 NIntegrate[f[x] Sin[n π x], {x, 0, 1}], {n, 1, nmax}]
```
tabela integra numérica··· seno

bnCoefs[g_, nmax_ : 100] := Table$\left[\frac{2}{n\,\pi}\right.$ NIntegrate[g[x] Sin[n π x], {x, 0, 1}], {n, 1, nmax}$\left.\right]$

tabela integra numérica··· seno

Example

```
In[ ]:= f[x_] := x (x^3 - 2 x^2 + 1);
g[x_] := x (1 - x);
an = anCoefs[f, 100] // Quiet;
                        silencioso
bn = bnCoefs[g, 100] // Quiet;
                        silencioso

(* I use this function "Compile" to make the evaluation of u faster *)
 unidade imaginária     compila
u = Compile[{{x, _Real}, {t, _Real}},
   compila
  Total@Table[(an[[n]] Cos[n π t] + bn[[n]] Sin[n π t]) Sin[n π x], {n, 1, Length[an]}]
        tabela      cosseno              seno          seno                  comprimento
 ];

In[ ]:= Animate[Plot[u[x, t], {x, 0, 1},
        anima   gráfico
   PlotRange → {-0.35, 0.35}
   intervalo do gráfico
  ], {t, 0, 3},
  AnimationRate → 0.3,
  velocidade da animação
  AnimationRepetitions → 1, AnimationRunning → False]
  repetições de animação     animação ativa       falso
```

Animate can be very heavy. I recommend you use it just for basic tests and then, to get better plots, export the outcomes as a GIF.

How to export as a GIF

The best way to export as a GIF is to create a list of plots and then export that list.

```
In[ ]:= plots = Table[Plot[u[x, t], {x, 0, 1},
                tabela gráfico
     PlotRange → {-0.35, 0.35},
     intervalo do gráfico
     FrameLabel → {"x", "u"}
     legenda do quadro
    ], {t, linspace[0, 3, 30]}];

••• Table: Iterator {t, linspace[0, 3, 30]} does not have appropriate bounds.

In[ ]:= Export["PDEs_wave1d_example.gif", plots, ImageResolution → 150];
        exporta                                   resolução da imagem
```

Example gallery

Plucked string

In[ ]:=
```
a = 0.7;
f[x_] := Piecewise[{{x / a, 0 ≤ x ≤ a}, {(1 - x) / (1 - a), a ≤ x ≤ 1}}];
         função por partes
g[x_] := 0;
an = anCoefs[f, 100] // Quiet;
                        silencioso
bn = bnCoefs[g, 100] // Quiet;
                        silencioso
u = Compile[{{x, _Real}, {t, _Real}},
    compila
   Total@Table[(an[[n]] Cos[n π t] + bn[[n]] Sin[n π t]) Sin[n π x], {n, 1, Length[an]}]
         tabela        cosseno           seno           seno                    comprimento
  ];
```

In[ ]:=
```
plots = Table[Plot[u[x, t], {x, 0, 1},
        tabela gráfico
     PlotRange → {-1, 1},
     intervalo do gráfico
     FrameLabel → {"x", "u"}
     legenda do quadro
   ], {t, N@Subdivide[0, 3, 10]}];
         v·· subdivide
Export["PDEs_wave1d_plucked.gif", plots];
exporta
```

In[ ]:=
```
a = 0.7;
f[x_] := Piecewise[{{x / a, 0 ≤ x ≤ a}, {(1 - x) / (1 - a), a ≤ x ≤ 1}}];
         função por partes
F = Interpolation[Table[{x, f[x]}, {x, 0, 1, 0.1}], InterpolationOrder → 3]
    interpolação  tabela                            ordem de interpolação
```

Out[ ]=
```
InterpolatingFunction[ Domain: {{0., 1.}}
                       Output: scalar ]
```

```
In[ ]:= InterpolatingFunction[ Domain: {{0., 1.}} Output: scalar ]
        função de interpolação

Out[ ]= InterpolatingFunction[ Domain: {{0., 1.}} Output: scalar ]
```

Gaussian wave-packet

```
In[ ]:= x0 = 0.5;
σ = 0.02;
f[x_] := Exp[-(x - x0)^2 / (2 σ^2)]
         exponencial
g[x_] := 0;
an = anCoefs[f, 100] // Quiet;
                        silencioso
bn = bnCoefs[g, 100] // Quiet;
                        silencioso
u = Compile[{{x, _Real}, {t, _Real}},
    compila
   Total@Table[(an⟦n⟧ Cos[n π t] + bn⟦n⟧ Sin[n π t]) Sin[n π x], {n, 1, Length[an]}]
         tabela      cosseno           seno          seno                comprimento
  ];

In[ ]:= plots = Table[Plot[u[x, t], {x, 0, 1},
                tabela gráfico
     PlotRange → {-1, 1},
     intervalo do gráfico
     FrameLabel → {"x", "u"}
     legenda do quadro
    ], {t, N@Subdivide[0, 3, 50]}];
           v·· subdivide
Export["PDEs_wave1d_gaussian.gif", plots];
exporta
```

URLs para acessar esse notebook

https://www.wolframcloud.com/obj/gtlandi/Published/WaveEquation_Annimated_GIFs.nb

Embedding:
<iframe src="https://www.wolframcloud.com/obj/gtlandi/Published/WaveEquation_Annimated_GIF-s.nb?_embed=iframe" width="600" height="800"></iframe>

Using Fourier Transforms for image compression

This code is based on:
https://demonstrations.wolfram.com/ImageCompressionViaTheFourierTransform/#popup1
by Chris Maes (March 2011)

The idea

1. Extracting image data

2. Viewing the image data as a sequence

3. Spectral analysis of the image (Fourier Transform)

4. Same thing, but now in 2D

Function implementation

In[ ]:=
```
ImageBW[img_] := N@Map[255 - Mean[#] &, 255 ImageData[img], {2}]
                  v··  aplica-s···  média          dados de imagem

CompressImage[IMG_, imsize_ : 300] := Manipulate[
                                      manipula
  Module[{cf, if},
  módulo de código
   cf = Chop[fimg, 10^Chop[c]];
        substitui números··  substitui números pequenos por 0
   if = InverseFourier[cf];
        transformada de Fourier discreta inversa
   Grid[{{
   grade
      Text@Row[{Style["original (", "Label"],
      texto    linha  estilo            etiqueta
         IntegerPart[Times @@ Dimensions[IMGBW] / 10^3.], " kb)"}], Text@Row[{
                     multiplica···  dimensões                         texto   linha
```

```
        Style["compressed (", "Label"],
                               etiqueta
        IntegerPart[((Times @@ Dimensions[IMGBW]) - Count[cf, 0, {2}]) / 10^3.],
                      multiplica ··· dimensões              contagem
        " kb; ",
                          ((Times @@ Dimensions[IMGBW]) - Count[cf, 0, {2}])
        IntegerPart[100. ----------------------------------------------------],
        parte inteira                  Times @@ Dimensions[IMGBW]

        "% of original size)"

       }]
    },
    {plot,
     ArrayPlot[if, ImageSize → imsize, PlotRangePadding → None, Frame → False]},
     represen··      tamanho da imagem      preenchimento de interva·· nen···  quadro   falso
    {Text@Style["error", "Label"], Text@Style["Power spectrum", "Label"]},
     texto  estilo             etiqueta  texto  estilo  potência              etiqueta
    {ArrayPlot[Abs[IMGBW - if], ImageSize → imsize, PlotRangePadding → None,
     representa··· valor absoluto  tamanho da imagem    preenchimento de interva·· nenhum
      Frame → False], ArrayPlot[RotateRight[Sqrt[Sqrt[Abs[cf]]], {128, 128}],
              falso    representa··· gira para direita raiz·· raiz·· valor absoluto
      ImageSize → imsize, PlotRangePadding → None, Frame → False]}}]
                          preenchimento de interva·· nen···  quadro   falso
 ],
 {{c, 1, "compression"}, -3, 3, .1}, Initialization :→ {
                                     inicialização
   IMGBW = ImageBW[IMG];
   plot :=
    ArrayPlot[IMGBW, ImageSize → imsize, PlotRangePadding → None, Frame → False];
    representação de arr··· tamanho da imagem  preenchimento de interva·· nen···  quadro  falso
   fimg := Fourier[N[IMGBW]];
           transfo··· valor numérico
  }, SynchronousInitialization → False]
     inicialização sincronizada    falso
```

The function **"CompressImage[img]"** creates a Manipulate object that allows you to play with different compression ratios.

The function also has the image size as an optional argument: CompressImage[img, imgsize].

Example gallery (include your own fun pics!)

Tip: don’t leave too many “Manipulate” objects open at the same time.
It makes the notebook very heavy.

Example 1

In[ ]:=
```
img = ExampleData[{"TestImage", "Lena"}]
```
dados de exemplo

Out[ ]=

```
(* Click me! *)
CompressImage[img]
```

Example 2

Example 3

In[ ]:= CompressImage[, 500]

Polinomios de langrange

In[ ]:=
$$G = \frac{1}{\sqrt{1 - 2\,x\,\lambda + \lambda^2}};$$

```
SeriesCoefficient[G, {λ, 0, 2}]
```
coeficiente de série

Out[ ]= $\frac{1}{2}\left(-1 + 3\,x^2\right)$

In[ ]:=
```
LegendreP[2, x]
```
polinómios de Legendre

Out[ ]= $\frac{1}{2}\left(-1 + 3\,x^2\right)$

In[ ]:=
```
Series[G, {λ, 0, 4}]
```
série

Out[ ]= $1 + x\,\lambda + \frac{1}{2}\left(-1 + 3\,x^2\right)\lambda^2 + \frac{1}{2}\left(-3\,x + 5\,x^3\right)\lambda^3 + \frac{1}{8}\left(3 - 30\,x^2 + 35\,x^4\right)\lambda^4 + O[\lambda]^5$

In[ ]:=
```
Table[LegendreP[n, x], {n, 0, 4}]
```
tabela | polinómios de Legendre

Out[ ]= $\left\{1,\ x,\ \frac{1}{2}\left(-1 + 3\,x^2\right),\ \frac{1}{2}\left(-3\,x + 5\,x^3\right),\ \frac{1}{8}\left(3 - 30\,x^2 + 35\,x^4\right)\right\}$

Visualizing the spherical harmonics

Just run the code below and play with the possible values of *l* and *m*.

In[ ]:=
```
Clear[l, m, θ, ϕ]
apaga
Manipulate[
manipula
 spheri = SphericalHarmonicY[l, m, θ, ϕ];
          função Y harmônico esférico
 ParametricPlot3D[
 gráfico paramétrico 3D
  Evaluate[{Sin[θ] Cos[ϕ], Sin[θ] Sin[ϕ], Cos[θ]} Abs[spheri]], {θ, 0, Pi},
  calcula   seno   cosseno seno  seno  cosseno valor absoluto      número pi
  {ϕ, -Pi, Pi}, PlotRange → {{-.5, .5}, {-.5, .5}, {-1.1, 1.1}},
      n⋯  n⋯   intervalo do gráfico
  Mesh → False,
  malha  falso
  PlotPoints → {36, 18},
  número de pontos no gráfico
  MaxRecursion → ControlActive[0, 2],
  recursão máxima  controle ativo
  ViewAngle → .246,
  ângulo de vista
  ImageSize → {500, 377},
  tamanho da imagem
  Axes → False,
         falso
  SphericalRegion → True, Boxed → False,
  re⋯                verd⋯ rodeado⋯ falso
  ColorFunctionScaling → False,
                         falso
  ColorFunction → Function[{x, y, z, θ, ϕ},
                  função
    Evaluate@If[Re@spheri > 0, orange, blue]],
             se parte real
  PlotLabel → Style[With[{l = l, m = Round@m},
              estilo com        arredondamento
     TraditionalForm[HoldForm[r == SphericalHarmonicY[l, m, θ, ϕ]]]], 14]],
                     forma sem avali⋯ função Y harmônico esférico
 {{l, 2, "degree l"}, 0, 7, 1, ControlType → Setter},
                               tipo de controle  seletor
 {{m, 0, "order m"}, -l, l, 1, Appearance → "Labeled"}]
                               aparência     etiquetado
```