`pdefourier` user manual

Emmanuel Roque, José Antonio Vallejo

April 22, 2020

Contents

1 Introduction **1**

2 Piecewise-defined functions **3**
2.1 Examples 5

3 Computation of Fourier coefficients **6**
3.1 Examples 7

4 Expansion of the coefficients **8**
4.1 Examples: 9

5 Frequency spectrum and harmonics **12**
5.1 Examples 12

6 Symbolic solution of PDEs **14**

7 Solving PDEs with `pdefourier` **18**
7.1 Heat equation 18
7.2 Laplace's equation 19
7.3 Wave equation 21

Bibliography **23**

1 Introduction

This Maxima package computes symbolically the Fourier of piecewise-smooth functions (available at `https://github.com/emmanuelroque/pdefourier`). Using the method of separation of variables it is also able to symbolically solve the one-dimensional heat and wave equations on a domain $[0, L]$, with regular Sturm-Liouville conditions, that is, general boundary conditions of the form:

$$\begin{aligned}\alpha_1 u(0,t) + \beta_1 u_x(0,t) &= h_1(t)\\ \alpha_2 u(L,t) + \beta_2 u_x(L,t) &= h_2(t)\end{aligned}$$

Moreover, the package can solve the two-dimensional Laplace equation for a variety of domains (rectangles, disks, annuli, wedges) either with Dirichlet or Neumann boundary conditions. In the case of a rectangular domain [0,a] x [0,b], the package can solve the Laplace equation with mixed boundary conditions of the form:

$$\begin{aligned}(1-\alpha)u(x,0) + \alpha u_y(x,0) &= f_0(x) \quad 0 \le x \le a\\ (1-\beta)u(x,b) + \beta u_y(x,b) &= f_b(x) \quad 0 \le x \le a\\ (1-\gamma)u(0,y) + \gamma u_x(0,y) &= g_0(y) \quad 0 \le y \le b\\ (1-\delta)u(a,y) + \delta u_x(a,y) &= g_a(y) \quad 0 \le y \le b\end{aligned}$$

where $\alpha, \beta, \gamma, \delta \in \{0,1\}$.

Of course, in all cases it is possible to truncate a series to make numerical calculations.

Remark 1.1. `pdefourier` automatically loads other packages, including `piecewise`, `draw`, `simplify_sum`, and `syntactic_factor`. The Maxima package `syntactic_factor` was written by Stavros Macrackis and released under the terms of LGPLv3.

Remark 1.2. There is a Maxima session in the documentation folder (in wxm format) with lots of examples, graphics, animations and tips for use. Here, we only give a quick introduction to the main commands used for solving typical problems.

2 Piecewise-defined functions

The package piecewise translates piecewise-defined functions to lists and viceversa. For this purpose, two functions were written: pw2list and list2pw. The function pw2list requires that a piecewise-defined expression is written in one of the following ways:

-(Bounded interval) if x>=a$_0$ and x<=a$_1$ then expr$_1$ elseif x>a$_1$ and x<= a$_2$ then expr$_2$ ... elseif x>a$_n$ and x<=a$_{n+1}$ then expr$_{n+1}$
-(Unbounded interval) if x<=a$_0$ then expr$_0$ elseif x>a$_0$ and x<=a$_1$ then expr$_1$ ... elseif x>=a$_n$ then expr$_n$
-(Left-bounded interval) if x>=a$_0$ and x<=a$_1$ then expr$_1$ elseif x>a$_1$ and x<= a$_2$ then expr$_2$ ... elseif x>a$_n$ then expr$_{n+1}$
-(Right-bounded interval) if x<=a$_1$ then expr$_1$ elseif x>a$_1$ and x<= a$_2$ then expr$_2$ ... elseif x>a$_n$ and x<=a$_{n+1}$ then expr$_{n+1}$

In each case, it is mandatory that $a_i <= a_{i+1}, a_i \in \mathbb{R}$ to avoid potential issues with the detection of valid intervals in all the equivalent ways of writing them, and do it efficiently. For instance, translating this mathematical sentence $x \in [a_0, a_1]$ into a valid Maxima's expression could be done in eight different but equivalent ways (see Table 1).

$a_0 <= x$ and $x <= a_1$	$x <= a_1$ and $a_0 <= x$
$a_0 <= x$ and $a_1 >= x$	$a_1 >= x$ and $a_0 <= x$
$x >= a_0$ and $x <= a_1$	$x <= a_1$ and $x >= a_0$
$x >= a_0$ and $a_1 >= x$	$a_1 >= x$ and $x >= a_0$

Table 1: Equivalent expressions for $x \in [a_0, a_1]$

Every expression in Maxima has an internal representation and the user can get each part of it using the command inpart. By imposing the restriction $a_i <= a_{i+1}$ as mentioned above, there are going to be only four equivalences (the ones appearing in the first column of the table) and those can be reduced inside pw2list to one using pattern matching and user-defined rules to only work with expressions of the form $a_i <= x$ and $x <= a_{i+1}$ and those obtained by replacing $<=$ with $<$ in one or both inequalities. The advantage of working only with these is that each endpoint of an interval a_i will have a "fixed position" in the internal representation of the conditional expression, making the task of translating it to a list and detecting empty intervals (e.g. $x < 0$ and $x >= 1$) easier. The output returned by pw2list is of the form:

$$\left[\left[\left[a_0, a_1\right], \text{expr}_1\right], \left[\left[a_1, a_2\right], \text{expr}_2, \right] \ldots\right] \tag{$\star$}$$

It is important to mention that currently the function list2pw_expr only deals with the bounded domain case.

The package also includes other functions to make operations between piecewise-defined functions such as the sum, difference and product of two piecewise expressions written as lists and functions to add or multiply a piecewise expression by a global factor. Lastly, there are functions to compute the derivative and the integral of a piecewise expression written as a list, all of these only work in the bounded domain case. Table 2 can be used as a reference. All lists are returned in $\star$ format. In the table, `pwexpr` refers to an expression of the form `if-else` corresponding to the bounded case described above and `pwlist` to a list as in the $\star$ format.

Command	Output
`pw2list(pwexpr, var)`	A `pwlist` in the variable `var` corresponding to `pwexpr`.
`list2pw_expr(pwlist, var)`	A `pwexpr` in the variable `var` corresponding to `pwlist`.
`pwsuml(Lf,Lg,var)`	A `pwlist` corresponding to the sum of the `pwlist` Lf and `pwlist` Lg in the variable `var`.
`pwdifferencel(Lf,Lg,var)`	A `pwlist` corresponding to the difference of the `pwlist` Lf and `pwlist` Lg in the variable `var`.
`pwprodl(Lf,Lg,var)`	A `pwlist` corresponding to the product of `pwlist` Lf and `pwlist` Lg in the variable `var`.
`pwdiffl(Lf,var,n)`	A `pwlist` corresponding to the n-th derivative of `pwlist` Lf with respect to variable `var`.
`pwintegratel(Lf,var)`	The integral of `pwlist` Lf with respect to variable `var`.
`pwglobalproductl(Lf, expr)`	A `pwlist` corresponding to the multiplication of `pwlist` Lf by global factor `expr`.
`pwglobalsuml(Lf, expr)`	A `pwlist` corresponding to the sum of `pwlist` Lf and global `expr`.

Table 2: Functions to work with piecewise-defined functions.

2.1 Examples

Here, we will show some examples of usage, the syntax is pretty intuitive. First, we need to load the package `pdefourier` that automatically loads a separate package called `piecewise`, which contains the functions `pw2list` and `list2pw` among others. Of course the package `piecewise` can be used independently but in this work we will focus on its joint functionality with `pdefourier`.

```
(%i1) load(pdefourier.mac)$
(%i2) u(x):=if (x>=-1 and x<-1/2) then -1-x
elseif (x>=-1/2 and x<1/2) then x^3
elseif (x>=1/2 and x<=1) then 1-x$
(%i3) absolute1(x):=if (x>=-1 and x<0) then -x
elseif (x>=0 and x<=1) then x$
(%i4) ul:pw2list(u(x),x);
(ul) [[[-1,-1/2],-x-1],[[-1/2,1/2],x^3],[[1/2,1],1-x]]
(%i5) absl:pw2list(absolute1(x),x);
(absl) [[[-1,0],-x],[[0,1],x]]
(%i6) pwsuml(ul,absl,x);
(%o6) [[[-1,-1/2],-2*x-1],[[-1/2,0],x^3-x],[[0,1/2],x^3+x],
        [[1/2,1],1]]
(%i7) pwdifferencel(ul,absl,x);
(%o7) [[[-1,-1/2],-1],[[-1/2,0],x^3+x],[[0,1/2],x^3-x],
        [[1/2,1],1-2*x]]
(%i8) pwglobalsuml(ul,%e);
(%o8) [[[-1,-1/2],-x+%e-1],[[-1/2,1/2],x^3+%e],
        [[1/2,1],-x+%e+1]]
(%i9) pwglobalproductl(ul,%e);
(%o9) [[[-1,-1/2],%e*(-x-1)],[[-1/2,1/2],%e*x^3],
        [[1/2,1],%e*(1-x)]]
(%i10) pwprodl(ul,absl,x);
(%o10) [[[-1,-1/2],-(-x-1)*x],[[-1/2,0],-x^4],
        [[0,1/2],x^4],[[1/2,1],(1-x)*x]]
(%i11) pwdiffl(ul,x,2);
(%o11) [[[-1,-1/2],0],[[-1/2,1/2],6*x],[[1/2,1],0]]
(%i12) pwintegrate(ul,x);
(%o12) 0
(%i13) pwintegrate(absl,x);
(%o13) 1
(%i14) list2pw_expr(absl,x);
(%o14) if -1<=x and x<0 then -x elseif 0<=x and x<=1 then x
(%i15) list2pw(absl,absolute,x);
(%o15) absolute(x):=if -1<=x and x<=0 then -x
        elseif 0<=x and x<=1 then x
```

3 Computation of Fourier coefficients

The strategy followed in our implementation of the package is to transform any trigonometric function into its canonical form internally so that it becomes easy to decide whether or not the input contains an expression whose Fourier coefficients must be computed separately to avoid potencial issues with trigonometric integrals using Maxima's pattern matching capabilities. We search for the following patterns:

$$x^r\cos(\frac{m\pi x}{L}), x^r\sin(\frac{m\pi x}{L}), \cos(\frac{m\pi x}{L}), \sin(\frac{m\pi x}{L}), \quad r, m \in \mathbb{N}$$

To do so, we used the Maxima built-in commands `defmatch` and `matchdeclare`.

```
/*Match rules for special cases*/

matchdeclare(nexp,lambda([e],e#0 and nonnegintegerp(e)))$
matchdeclare(nfreq,lambda([e],e#0 and nonnegintegerp(e)))$
matchdeclare(const,lambda([e],e#0 and freeof(xargument,e)))$

defmatch(powxsin,
const*xargument^nexp*sin(nfreq*%pi*xargument/pargument),
xargument,pargument)$
defmatch(powxcos,
const*xargument^nexp*cos(nfreq*%pi*xargument/pargument),
xargument,pargument)$
defmatch(multsin,const*sin(nfreq*%pi*xargument/pargument),
xargument,pargument)$
defmatch(multcos,const*cos(nfreq*%pi*xargument/pargument),
xargument,pargument)$
```

We transform the input expression $\text{term}_1 + \ldots + \text{term}_s$ to a list of its terms $[\text{term}_1, \ldots, \text{term}_s]$ and search one by one for the special patterns to compute the coefficients correctly. The returned answer is of the form $[[a_0, a_n, b_n], \text{l.s.v}]$ where l.s.v is an abbreviation for *list of singular values*, meaning all the values of a_n, b_n that must be computed separately. The l.s.v is written in the format $[[j, a_j, b_j], \ldots]$. Similar ideas also apply for the case of the complex, sine and cosine coefficients, the only difference is in the way the answer is given (see Table 3).

Type	Command	Answer format	l.s.v format
Trigonometric	`fouriercoeff(expr, var, L)`	$[[a_0, a_n, b_n], \text{l.s.v}]$	$[]$ / $[[j, a_j, b_j], \ldots]$
Complex	`cfouriercoeff(expr, var, L)`	$[[c_0, c_n], \text{l.s.v}]$	$[]$ / $[[j, c_j], \ldots]$
Cosine	`fouriercoscoeff(expr, var, L)`	$[[a_0, a_n], \text{l.s.v}]$	$[]$ / $[[j, a_j], \ldots]$
Sine	`fouriersincoeff(expr, var, L)`	$[[b_n], \text{l.s.v}]$	$[]$ / $[[j, b_j], \ldots]$

Table 3: Output formats for the Fourier coefficients (l.s.v=list of singular values).

3.1 Examples

```
(%i5) fouriercoeff((3*x^2*cos(7*x)+x),x,%pi);
```

$$[[-\frac{6}{49},\frac{12(n^2+49)(-1)^{n+1}}{n^4-98n^2+2401},\frac{2(-1)^{n+1}}{n}],[[7,\frac{98\pi^2+3}{98},\frac{2}{7}]]] \qquad (\%o5)$$

```
(%i6) fouriercoscoeff(cos(x)^2,x,%pi);
```

$$[[\frac{1}{2},0],[[2,\frac{1}{2}]]] \qquad (\%\ o6)$$

```
(% i7)  fouriercoscoeff((1+cos(2*x))/2,x,%pi);
```

$$[[\frac{1}{2},0],[[2,\frac{1}{2}]]] \qquad (\%\ o7)$$

Here we have an example of a piecewise-defined function having singular values. We will use the command `cfouriercoeff` to get the complex Fourier coefficients so that we can compare our answer to the one returned by Mathematica$^{\mathrm{TM}}$.

```
(% i8) f(x):=if x>=-%pi and x< 0 then x
elseif x>=0 and x<=%pi then sin(3*x)$
(% i9) cfouriercoeff(f(x),x,%pi);
```

$$[[-\frac{3\pi^2-4}{12\pi},\frac{\left(\%i\pi n^3-4n^2-9\%i\pi n+9\right)\,(-1)^n-2n^2-9}{2\pi n^2\,(n^2-9)}],[[3,-\frac{15\%i\pi-4}{36\pi}]]]$$

However, Mathematica $^{\mathrm{TM}}$ is not able to detect the singular value of the coefficient when $n=3$.

```
In[8]:=f[x]:= Piecewise[{{x,-Pi<x<0}, {Sin[3x],0<x<Pi}}]
In[9]:=FourierCoefficient[f[x], x, n]
```

$$\begin{cases} \frac{1}{3\pi}-\frac{\pi}{4} & n=0 \\ \frac{(-1)^n\left(i\pi n^3-2((-1)^n+2)n^2-9i\pi n-9(-1)^n+9\right)}{2\pi n^2(n^2-9)} & \text{True} \end{cases}$$

Expansion routine	Series
`fouriercoeff_expand(lcoeff,var,L,N)`	`fourier_series(expr,var,L,N)`
`cfouriercoeff_expand(lcoeff,var,L,N)`	`cfourier_series(expr,var,L,N)`
`fouriersincoeff_expand(lcoeff,var,L,N)`	`fouriersin_series(expr,var,L,N)`
`fouriercoscoeff_expand(lcoeff,var,L,N)`	`fouriercos_series(expr,var,L,N)`

Table 4: Expansion routines and Fourier series syntax in `pdefourier`. Here `lcoeff`=list of coefficients.

4 Expansion of the coefficients

Fourier series are obtained from an expansion routine of the Fourier coefficients. The upper limit of summation can be a positive integer or infinite. In the first case, a truncated series is returned; in the second, a symbolic series is displayed. In Table 4 we summarize the syntax for the expansion routines along with the different Fourier series commands, see also Table 3 for comparison.

Now, let us see an example of how Fourier series are displayed symbolically. If $g(x) = x^4$ on $[-\pi, \pi]$ we get an answer like the ones appearing in Fourier Analysis textbooks.

```
(% i7) fourier_series(x^4,x,%pi,inf);
```

$$8\left(\sum_{n=1}^{\infty} \frac{\left(\pi^2\, n^2 - 6\right)\, (-1)^n \cos\left(n x\right)}{n^4}\right) + \frac{\pi^4}{5} \tag{\% o7}$$

It is important to notice that displaying infinite series correctly has been a source of troubles in different CAS. This is mainly due to the fact that they are not evaluated, only displayed symbolically, and so, a very human simplification might not be performed by a CAS. For illustration purposes, suppose that we want to compute the Fourier series of $h(x) = \sin 15x$ on the interval $[-\pi, \pi]$. It is obvious that its Fourier series is exactly equal to $h(x)$, because its Fourier coefficients are $a_0 = 0$, $a_n = 0$, $b_n = \delta_n^{15}$ and we have:

$$h(x) \sim \sum_{n=1}^{\infty} \delta_n^{15} \sin nx = \sin 15x$$

Following our approach, we avoid the issue of the evaluation of Kronecker delta functions inside an infinite sum:

```
(% i10)fourier_series(sin(15*x),x,%pi,inf);
```

$$\sin\left(15 x\right) \tag{\% o10}$$

When the list of singular values is not empty, we have to print a message indicating the set of indices that are taken into account in the infinite sum. We show an example of how to use the expansion routines to obtain the Fourier series, truncated or not, and how they handle singular values in the Fourier coefficients when displaying an infinite series.

```
(% i11) fcoeff:fouriersincoeff(x*cos(3*x),x,%pi);
```

$$[[\frac{2n\left(-1\right)^{n}}{n^{2}-9}],[[3,-\frac{1}{6}]]] \tag{fcoeff}$$

```
(% i12) fouriersincoeff_expand(fcoeff,x,%pi,5);
```

$$-\frac{5\sin\left(5x\right)}{8}+\frac{8\sin\left(4x\right)}{7}-\frac{\sin\left(3x\right)}{6}-\frac{4\sin\left(2x\right)}{5}+\frac{\sin\left(x\right)}{4} \tag{\% o12}$$

```
(% i13) fouriersincoeff_expand(fcoeff,x,%pi,inf);
```

The sum is over \N- {3}

$$2\left(\sum_{n=1}^{\infty}\frac{n\left(-1\right)^{n}\sin\left(nx\right)}{n^{2}-9}\right)-\frac{\sin\left(3x\right)}{6} \tag{\% o13}$$

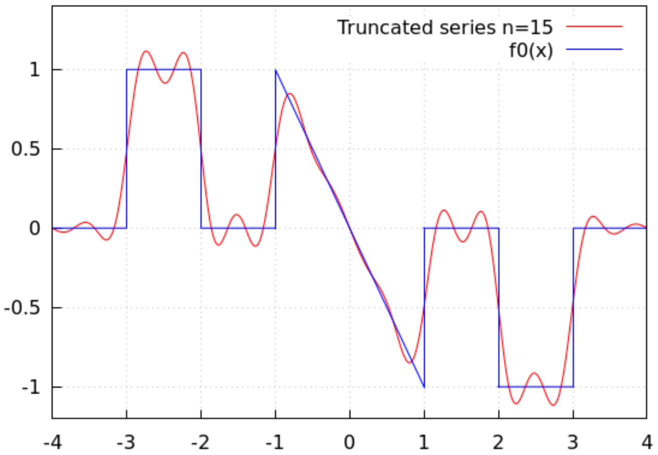
In case we are using the graphical user interface wxMaxima[1] , we can write a very simple function to see how well Fourier series approximates a function:

```
(%i3) wx_fourieraprox(fname,p,NN,y1,y2):=block(
    expr:rhs(apply(fundef,[fname])),
    ss:fourier_series(expr,x,p,NN),
    wxdraw2d(yrange=[y1,y2],grid=true,
    color=red,line_width=1,nticks=200,
            key=sconcat("Truncated series n=",string(NN)),
    explicit(ss,x,-p,p),
    color=blue,line_width=1,nticks=200,
            key=sconcat(string(fname),"(x)"),
            explicit(expr,x,-p,p))
)$
```

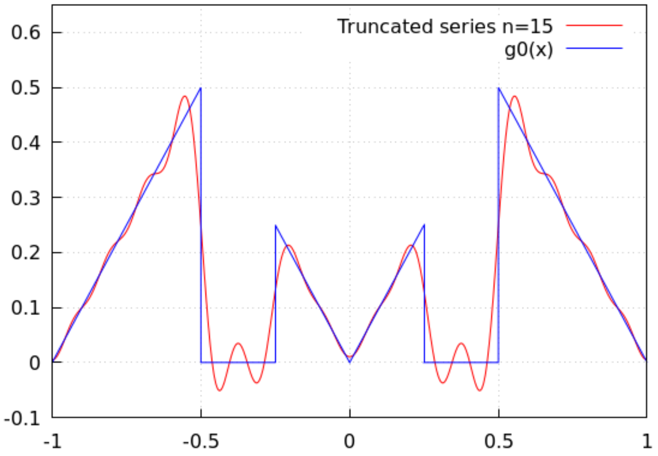
4.1 Examples:

```
(%i15) f0(x):=if (x>=-4 and x<-3 ) then 0
        elseif (-3<=x and x<=-2) then 1
        elseif (-2<x and x<-1) then 0
elseif (-1<=x and x<=1) then -x
        elseif (1<=x and x<=2) then 0
elseif (2<x and x<3) then -1 elseif (x>=3 and x<=4) then 0$
(%i16) wx_fourieraprox(f0,4,15,-1.2,1.4);
(%t16)  (Graphics)
(%o16)
```

[1] wxMaxima, a GUI for Maxima CAS, https://wxmaxima-developers.github.io/wxmaxima/

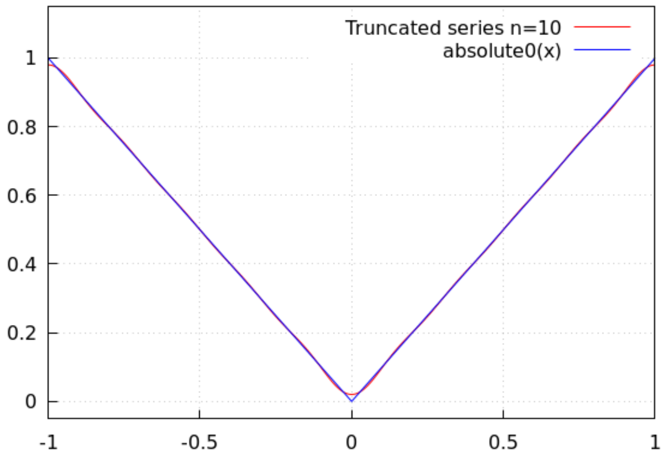


```
(%i17) g0(x):=if (x>=-1 and x<-1/2  ) then x+1
elseif (x>=-1/2 and x<-1/4) then 0
elseif (x>=-1/4 and x<=0) then -x
elseif (x>=0 and x<=1/4) then x
elseif (x>=1/4 and x<1/2) then 0
elseif (x>=1/2 and x<=1) then 1-x$
(%i18) wx_fourieraprox(g0,1,15,-0.1,0.65);
(%t18)  (Graphics)
(%o18)
```

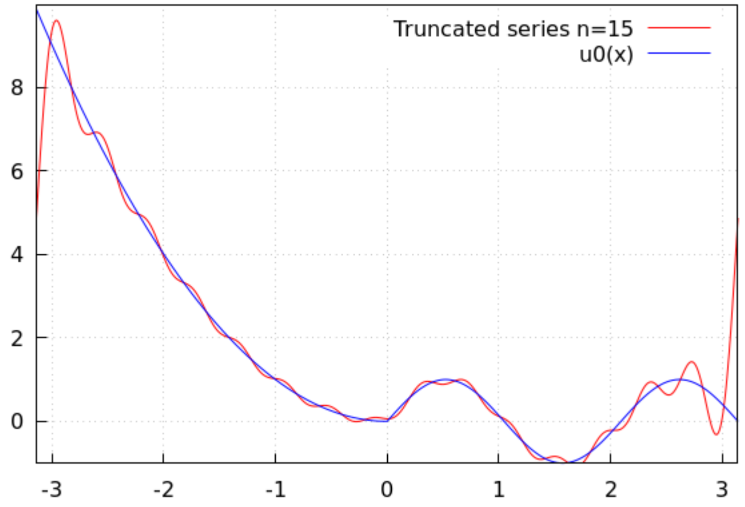


```
(%i19) absolute0(x):=if ( x>=-1 and x<=0) then -x
        elseif (x>0 and x<=1) then x$
```

```
(%i20) wx_fourieraprox(absolute0,1,10,-0.05,1.15);
(%t20)  (Graphics)
(%o20)
```



```
(%i21) u0(x):=if (x>=-%pi and x<0) then x^2
        elseif x>=0 and x<=%pi then sin(3*x)$
(%i22) wx_fourieraprox(u0,%pi,15,-1,%pi^2+1/10);
(%t22)  (Graphics)
(%o22)
```



5 Frequency spectrum and harmonics

Frequency analysis is very useful in Engineering applications (but also in Physics). This technique requires that the Fourier series be first re-expressed by using the identity:

$$a\cos(\omega) + b\sin(\omega) = r\cos(\omega - u)$$

where the modulus and the phase shift are given, respectively, by $r = \sqrt{a^2 + b^2}$ and $u = \arctan(b/a)$. We can rewrite the terms summed in the series as the so-called harmonics $c_n \cos(n\omega x - u_n)$.

The frequency analysis is done in `pdefourier` with the aid of the function `fourier_freq`. The function `fourier_harm` returns a list with the first n harmonics of a given function, with the syntax `fourier_harm(expr,var,L,n)` where $L = (b - a)/2$, and $[a, b]$ is the interval of definition of the function, while `fourier_freq_list` (with the same syntax) gives the list of the amplitudes.

5.1 Examples

We perform the computations on some previously defined functions in the last section examples.

```
(% i23) fourier_harm(g0(x),x,1,3);
```

$$[-\frac{\left(-\frac{\pi}{\sqrt{2}} + 2\pi - 2^{\frac{3}{2}}\right)\cos(\pi x)}{2\pi^2}, -\frac{(12 - 2\pi)\cos(2\pi x)}{8\pi^2}, \frac{\left(\frac{3\pi}{\sqrt{2}} + 6\pi - 2^{\frac{3}{2}}\right)\cos(3\pi x)}{18\pi^2}] \tag{\% o23}$$

```
(% i24) fourier_freq_list(g0(x),x,1,3);
```

$$[[1.0, 0.06248055460146331], [2.0, 0.07240430391755899], [3.0, 0.1276953872275108]] \tag{\% o24}$$

```
(% i25) fourier_harm(f0(x),x,4,5);
```

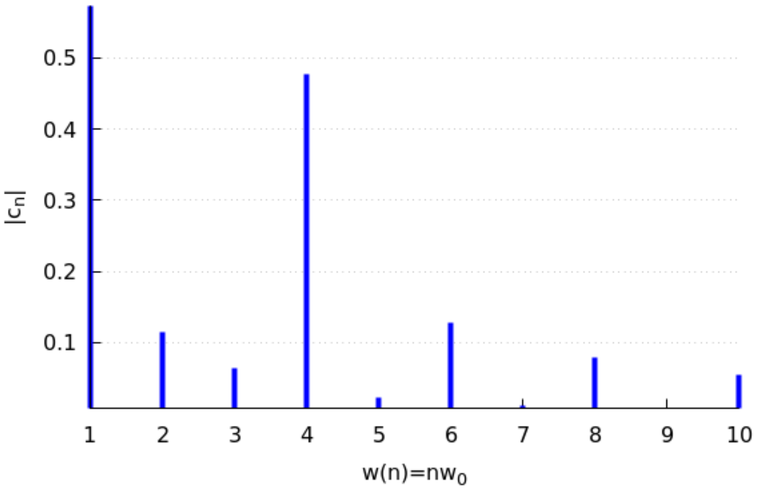
$$[-\frac{2^{\frac{5}{2}}\sin\left(\frac{\pi x}{4}\right)}{\pi^2}, \frac{(2\pi - 4)\sin\left(\frac{\pi x}{2}\right)}{2\pi^2}, -\frac{2^{\frac{5}{2}}\sin\left(\frac{3\pi x}{4}\right)}{9\pi^2}, -\frac{3\sin(\pi x)}{2\pi}, \frac{2^{\frac{5}{2}}\sin\left(\frac{5\pi x}{4}\right)}{25\pi^2}] \tag{\% o25}$$

```
(% i26) fourier_freq_list(f0(x),x,4,3);
```

$$[[1.0, 0.5731591682507566], [2.0, 0.1156675188991151], [3.0, 0.06368435202786184]] \tag{\% o26}$$

In case we are using the graphical user interface wxMaxima, we also provide the command `wxfourier_freq(expr,var, L, n)` to plot the first n frequencies.

```
(% i27) wxfourier_freq(f0(x), x, 4,10);
(% o27)
```



6 Symbolic solution of PDEs

In the previous section, it was mentioned that Fourier series are obtained by performing an expansion of the Fourier coefficients. This approach is multipurpose. We've already seen that it allows a user to expand a list of Fourier coefficients that have been computed previously, but it is also useful to code the solution of the PDEs in an easier way. Although the package can solve the heat equation with a heat source $Q(x,t)$, we want to show how the expansion routines facilitate the process of finding solutions with an easier version of the equation. Consider the IBVP:

$$\begin{cases} u_t = \kappa u_{xx} & (x,t) \in [0,L] \times [0,\infty) \\ u(0,t) = 0 & u(L,t) = 0 \\ u(x,0) = f(x) & \end{cases}$$

Then, the solution is given by

$$u(x,t) = \sum_{n=1}^{\infty} b_n \exp[-\kappa(\frac{\pi n}{L})^2 t] \sin\frac{n\pi x}{L}$$
$$b_n = \frac{2}{L}\int_0^L f(x) \sin\frac{n\pi x}{L}$$

Notice that for our purposes, it will be sufficient if we create a list of the form $[[c_n(t)],[\,]/[j,c_j(t)]]$ where $c_n(t) = b_n \exp[-\kappa(\frac{\pi n}{L})^2 t]$ and then use the expansion routine corresponding to a Fourier sine series on the interval $[0,L]$ and the space variable x. For instance, if we consider $f(x) = x^2(1-x)$, $L = 1$, $\kappa = 1$, a solution to the IBVP can be obtained in a few lines of code:

```
(%i12) f(x):=if (0<=x and x<=1) then x^2*(1-x)$
(%i13) L:1$
(%i14) [[bn],lsv]:fouriersincoeff(f(x),x,L)$
(%i15) cn:bn*exp(-n^2*%pi^2*t)$
(%i16) fouriersincoeff_expand([[cn],[]],x,L,inf);
(%o16)
```

$$-\frac{4\sum_{n=1}^{\infty}\frac{(2(-1)^n+1)\,\%e^{-\pi^2\,n^2\,t}\sin(\pi n x)}{n^3}}{\pi^3} \qquad (\%\ o16)$$

Here the list of singular values was empty (obviously), and no more work was required. Similar methods to obtain solutions for the three equations with different types of boundary conditions and domains were implemented, dealing with general expressions (piecewise defined or not) and taking care of the possible singular values in the coefficients. It is important to mention that currently, the package is only able to detect correctly singular values in the case of Dirichlet and Neumann conditions, because their corresponding Sturm-Liouville problems lead to the sine and cosine orthogonal systems, however, a general heuristic routine able to detect eigenfunctions of other orthogonal systems remains to be implemented.

Now we introduce the IBVPs that can be solved using `pdefourier`. The general problem for the heat and wave equations can be written as:

$$\begin{cases} \mathcal{L}u = Q(x,t) \\ u(x,0) = F(x) \\ u_t(x,0) = G(x) \text{ (Wave equation)} \\ \alpha_1 u(0,t) + \beta_1 u_x(0,t) = h_1(t) \\ \alpha_2 u(L,t) + \beta_2 u_x(L,t) = h_2(t) \end{cases} \tag{6.1}$$

where $\mathcal{L}$ denotes either the wave operator $\mathcal{L}_{wave} = \frac{\partial^2}{\partial t^2} - c^2 \frac{\partial^2}{\partial x^2}$ or the heat operator $\mathcal{L}_{heat} = \frac{\partial}{\partial t} - \kappa \frac{\partial^2}{\partial x^2}$. We are also able to solve the general parabolic equation with constant coefficients and initial and boundary conditions as above, since a change of variables reduces it to the heat equation.

$$u_t = \kappa u_x x + v u_x + cu + Q(x,t) \tag{6.2}$$

The general problem for the Laplace's equation in a rectangle, an annulus and a disk (or a wedge depending on α being less or equal to 2π) can be written as shown below. In the case of an annulus as a domain, the Laplace's equation is solved in polar coordinates as well, and it is only needed to add an extra condition on the inner boundary.

$$\begin{cases} \textbf{Rectangle} \\ u_{xx} + u_{yy} = 0 \quad (x,y) \in [0,a] \times [0,b] \\ \alpha, \beta, \gamma, \delta \in \{0,1\} \\ (1-\alpha)u(x,0) + \alpha u_y(x,0) = f_0(x) \quad 0 \le x \le a \\ (1-\beta)u(x,b) + \beta u_y(x,b) = f_b(x) \quad 0 \le x \le a \\ (1-\gamma)u(0,y) + \gamma u_x(0,y) = g_0(y) \quad 0 \le y \le b \\ (1-\delta)u(a,y) + \delta u_x(a,y) = g_a(y) \quad 0 \le y \le b \end{cases} \tag{6.3}$$

$$\begin{cases}\textbf{Disk/Wedge/Annulus}\\ u_{rr}+\frac{1}{r}u_r+\frac{1}{r^2}u_{\theta\theta}=0\\ (r,\theta)\in(0,R_1]\times[0,\alpha[,\ 0<\alpha\le 2\pi,\ 0<R_1 \text{ (Disk/Wedge)}\\ (r,\theta)\in[R_1,R_2]\times[0,2\pi[,\ 0<R_1<R_2 \text{ (Annulus)}\\ \text{Dirichlet boundary conditions}\\ u(r,0)=0=u(r,\alpha) \text{ if } \alpha<2\pi \text{ (Wedge)}\\ u(R_1,\theta)=f(\theta) \text{ (Annulus/Disk/Wedge)}\\ u(R_2,\theta)=g(\theta) \text{ (Annulus)}\\ \text{Neumann boundary conditions}\\ u(r,0)=0=u(r,\alpha) \text{ if } \alpha<2\pi \text{ (Wedge)}\\ u_r(R_1,\theta)=f(\theta) \text{ (Disk/Wedge)}\end{cases} \tag{6.4}$$

Lastly, it is also possible to solve the Poisson equation on a rectangle with Dirichlet boundary conditions.

$$\begin{cases}u_{xx}+u_{yy}=Q(x,y) \quad (x,y)\in[0,a]\times[0,b]\\ u(x,0)=f_0(x) \quad 0\le x\le a\\ u(x,b)=f_b(x) \quad 0\le x\le a\\ u(0,y)=g_0(y) \quad 0\le y\le b\\ u(a,y)=g_a(y) \quad 0\le y\le b\end{cases} \tag{6.5}$$

In Table 6 we present the syntax for all the commands available at the time of this writing in `pdefourier` to solve partial differential equations. The notation is in accordance to the one appearing in IBPVs 6.1, 6.2, 6.3, 6.4 and 6.5. In the table `x,y` stand for space variables and `t` for the time variable, these are only for reference and other symbols can be used as variables. In the particular case of the Laplace's equation in a disk, wedge or an annuli the variable is identified with θ in the table.

It is also important to notice that the commands for mixed conditions, in particular, can solve IBVPs with Dirichlet or Neumann conditions. For instance, `mixed_heat`$(Q(x,t),F(x),1,0,1,0,h_1(t),h_2(t),x,t,L,\kappa,N)$ is equivalent to `dirichlet_heat`$(Q(x,t),F(x),h_1(t),h_2(t),x,t,L,\kappa,N)$. We provide the commands for Dirichlet and Neumann conditions separately for convenience and practicality, as those boundary conditions are the most common ones.

Heat equation
`dirichlet_heat`$(Q(x,t), F(x), h_1(t), h_2(t), x, t, L, \kappa, ord)$
`neumann_heat`$(Q(x,t), F(x), h_1(t), h_2(t), x, t, \kappa, L, ord)$
`mixed_heat`$(Q(x,t), F(x), \alpha_1, \beta_1, \alpha_2, \beta_2, h_1(t), h_2(t), x, t, L, \kappa, ord)$
`mixed_parabolic`$(Q(x,t), F(x), \alpha_1, \beta_1, \alpha_2, \beta_2, h_1(t), h_2(t), x, t, L, \kappa, v, c, ord)$
Laplace's equation
`dirichlet_laplace_disk(`$R_1, f(\theta), \theta, ord)$
`neumann_laplace_disk(`$R_1, f(\theta), \theta, ord)$
`dirichlet_laplace_annulus(`$R_1, R_2, f(\theta), g(\theta), \theta, ord)$
`dirichlet_laplace_wedge(`$R_1, \alpha, f(\theta), \theta, ord)$
`neumann_laplace_wedge(`$R_1, \alpha, f(\theta), \theta, ord)$
`dirichlet_laplace_rectangle(`$a, b, f_0(x), f_b(x), g_0(y), g_a(y), x, y, ord)$
`neumann_laplace_rectangle(`$a, b, f_0(x), f_b(x), g_0(y), g_a(y), x, y, ord)$
`mixed_laplace_rectangle(`$\alpha, \beta, \gamma, \delta, f_0(x), f_b(x), g_0(x), g_a(x), a, b, x, y, ord)$
`dirichlet_poisson_rectangle`$(Q(x,y), a, b, f_0(x), f_b(x), g_0(y), g_a(y), x, y, ord)$
Wave equation
`dirichlet_wave(`$Q(x,t), f(x), g(t), h_1(t), h_2(t), x, t, L, c, ord)$
`neumann_wave((`$Q(x,t), f(x), g(x), h_1(t), h_2(t), x, t, L, c, ord)$
`mixed_wave((`$Q(x,t), F(x), G(x), \alpha_1, \beta_1, \alpha_2, \beta_2, h_1(t), h_2(t), x, t, L, c, ord)$

Table 6: Commands in pdefourier to solve PDEs. The variable ord is the upper limit of summation can be either a positive integer or inf.

7 Solving PDEs with `pdefourier`

We will show some examples for each equation. Most of the examples are textbook problems tested in Mathematica$^{\text{TM}}$and Maple$^{\text{TM}}$and reported in [1]. Although `pdefourier` is able to solve these equations with many more boundary conditions, the ones appearing next are meant to give a general idea about how to use the package to solve PDEs. In some cases the output has been slightly edited to fit within the page. See Table 6 for reference. The examples marked with ($\star$) can't be solved in Mathematica$^{\text{TM}}$12.0.0.0, however all of them can be solved in Mathematica$^{\text{TM}}$12.1.0.0. Maple 2020 is able to solve all of them.

7.1 Heat equation

Example 7.1 ($\star$)**.** Consider the Neumann problem:

$$\begin{cases} u_t = u_{xx} + 1 + x\cos(t) \qquad (x,t) \in [0,1] \times \mathbb{R}^+ \\ u_x(0,t) = \sin(t) \quad u_x(1,t) = \sin(t) \\ u(x,0) = 1 + \cos(2\pi x) \end{cases}$$

This is Problem 6.17 in [5]. The solution is given by:

```
(%i35) neumann_heat(1+x*cos(t),1+cos(2*%pi*x),sin(t),sin(t),
        x,t,1,1,inf);
(%o35)
```

$$\%e^{-4\pi^2 t}\cos(2\pi x) + \sin(t)x + t + 1$$

Example 7.2 ($\star$)**.** Consider the Dirichlet problem:

$$\begin{cases} u_t = \kappa u_{xx} + \kappa \qquad (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(0,t) = A \quad u(L,t) = B \quad A, B \in \mathbb{R} \\ u(x,0) = F(x) \end{cases}$$

This is Exercise 8.2.1(d) in [2]. The solution is given by:

```
(%i2) declare(F,real)$
(%i3) declare(L,constant)$
(%i4) assume(L>0)$
(%i5) dirichlet_heat(%kappa,F(x),A,B,x,t,L,%kappa,inf);
(%o5)
```

$$\frac{\sum_{n=1}^{\infty} \frac{\%e^{\frac{-\kappa\pi^2 n^2 t}{L^2}} \sin(\frac{n\pi x}{L})\left(2\pi^3 n^3 \int_0^L F(x)\sin(\frac{n\pi x}{L})dx + (2L^3 - 2L^3(-1)^n)\%e^{\frac{\pi^2 n^2 t}{L^2}}\right)}{n^3}}{\pi^3 L}$$

$$+\frac{\sum_{n=1}^{\infty} \frac{\%e^{\frac{-\kappa\pi^2 n^2 t}{L^2}} \sin(\frac{n\pi x}{L})\left((2\pi^2 LBn^2 + L^3)(-1)^n - 2\pi^2 LAn^2 - 2L^3\right)}{n^3}}{\pi^3 L} - \frac{(A-B)x}{L} + A$$

Notice that $F(x)$ was declared as real. This is necessary so that Maxima does not display the solution in terms of the imaginary and real part of F. Also, this is a good example that shows the symbolic capabilities of the package.

Example 7.3 ($\star$)**.** Consider the following problem with mixed boundary conditions:

$$\begin{cases} u_t = u_{xx} \qquad (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(0,t) + u_x(0,t) = 0 \\ u(L,t) + u_x(L,t) = 0 \\ u(x,0) = f(x) \end{cases}$$

The solution can be obtained with the following commands:

```
(%i19) assume(L>0)$
(%i20) assume(t>0)$
(%i21) Q(x,t):=0$
(%i22) h1(t):=0$
(%i23) h2(t):=0$
(%i24) mixed_heat(Q(x,t),f(x),1,1,1,1,h1(t),h2(t),x,t,L,k,inf);
(%o24)
```

$$2\sum_{n=1}^{\infty}\frac{\%e^{-\frac{\pi^2 k\, n^2 t}{L^2}}\left(\sin\left(\frac{\pi n x}{L}\right)+\frac{\pi n\cos\left(\frac{\pi n x}{L}\right)}{L}\right)\int_0^L \mathrm{f}(x)\left(L\sin\left(\frac{\pi n x}{L}\right)+\pi n\cos\left(\frac{\pi n x}{L}\right)\right)dx}{\pi^2\, n^2+L^2}$$

7.2 Laplace's equation

Example 7.4 ($\star$)**.** Here we consider the Laplace's equation on a wedge and Neumann conditions:

$$\begin{cases} u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 0 \quad (r,\theta) \in [0,R] \times [0,\alpha],\ 0 < \alpha < 2\pi \\ u(r,0) = 0 \quad u(r,\alpha) = 0 \\ u_r(R,\theta) = f(\theta) \end{cases}$$

Here is the solution:

```
(%i26) declare(f,real)$
(%i27) neumann_laplace_wedge(1,%pi/2,f(%theta),%theta,inf);
(%o27)
```

$$\frac{2\sum_{n=1}^{\infty}\frac{\sin(2\theta n)\int_0^{\frac{\pi}{2}}\ \mathrm{f}(\theta)\sin(2\theta n)d\theta\ r^{2n}}{n}}{\pi}$$

Example 7.5. Here we consider the Laplace's equation on a disk and Dirichlet conditions:

$$\begin{cases} u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 0 \quad (r,\theta) \in [0,1] \times [0,2\pi] \\ u(1,\theta) = \sin^2 5\theta + \cos 2\theta \end{cases}$$

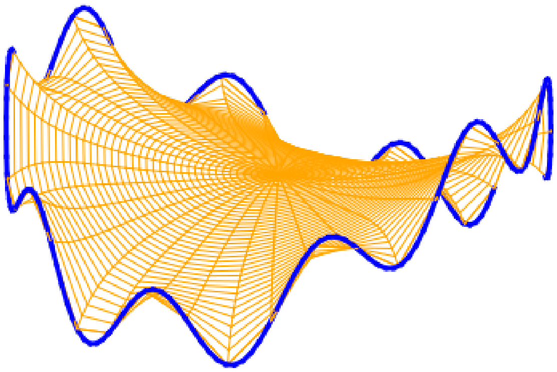
Then the solution is given by

```
(%i45) h(theta):=sin(5*theta)^2+cos(2*theta)$
(%i46) dirichlet_laplace_disk(1,h(theta),theta,inf)$
(%i47) expr2:ratsimp(demoivre(%));
```

$$-\frac{r^{10}\cos(10\theta)-2r^2\cos(2\theta)-1}{2} \tag{expr2}$$

We can plot the result as a parametric_surface.

```
wxdraw3d(view=[53,27],surface_hide=true, axis_3d=false,
color=orange, xu_grid=40,yv_grid=40,parametric_surface(
    r*cos(theta),r*sin(theta),expr2,r,0,1,theta,0,2*%pi
    ),
color=blue,line_width=2,nticks=200,
parametric(cos(theta),sin(theta),h(theta),theta,0,2*%pi));
```



Example 7.6. Here, we will show an example of the Laplace equation with mixed boundary conditions in a rectangle. The following problem is Exercise 2.5.1(a) in [2]

$$\begin{cases} u_{xx}+u_{yy}=0 \quad (x,y)\in[0,L]\times[0,H] \\ u(x,0)=0 \quad 0\le x\le L \\ u(x,b)=f(x) \quad 0\le x\le L \\ u_x(0,y)=0 \quad 0\le y\le H \\ u_x(a,y)=0 \quad 0\le y\le H \end{cases}$$

```
(%i2) declare(f,real)$
(%i3) assume(L>0)$
(%i4) assume(H>0)$
(%i5) mixed_laplace_rectangle(0,0,1,1,0,f(x),0,0,L,H,x,y,inf);
```

$$\frac{2\sum_{n=1}^{\infty}\frac{\cos\left(\frac{\pi n x}{L}\right)\int_0^L \mathrm{f}(x)\cos\left(\frac{\pi n x}{L}\right)dx\,\sinh\left(\frac{\pi n y}{L}\right)}{\sinh\left(\frac{\pi H n}{L}\right)}}{L}+\frac{\int_0^L \mathrm{f}(x)dx\,y}{HL}$$

Example 7.7 ($\star$)**.** This is Exercise 2.5.1(d) in [2]

$$\begin{cases} u_{xx} + u_{yy} = 0 \quad (x,y) \in [0,L] \times [0,H] \\ u_y(x,0) = 0 \quad 0 \le x \le L \\ u(x,H) = 0 \quad 0 \le x \le L \\ u(0,y) = g(y) \quad 0 \le y \le H \\ u(L,y) = 0 \quad 0 \le y \le H \end{cases}$$

```
(%i6) assume(L>0)$
(%i7) assume(H>0)$
(%i8) declare(g,real)$
(%i9) mixed_laplace_rectangle(1,0,0,0,0,0,g(y),0,L,H,x,y,inf);
```

$$\frac{2\sum_{n=1}^{\infty} \frac{\sinh\left(\frac{\pi(2n-1)\,(L-x)}{2H}\right)\cos\left(\frac{\pi(2n-1)y}{2H}\right)\int_0^H \mathrm{g}(y)\cos\left(\frac{(2\pi n-\pi)y}{2H}\right)dy}{\sinh\left(\frac{\pi L\,(2n-1)}{2H}\right)}}{H}$$

7.3 Wave equation

Example 7.8 ($\star$)**.** In the following IBVP we will consider the wave equation with homogeneous boundary conditions and a driving term. This is Exercise 8.5.2(b) in [2].

$$\begin{cases} u_{tt} = c^2 u_{xx} + \cos(\omega t) r(x) \quad (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(0,t) = 0 \quad u(L,t) = 0 \\ u(x,0) = f(x) \quad u_t(x,0) = 0 \end{cases}$$

Then:

```
(%i31) assume(L>0)$
(%i32) assume(t>0)$
(%i33) declare(r,real,f,real)$
(%i34) dirichlet_wave(cos(%omega*t)*r(x),f(x),0,0,0,x,t,L,c,inf);
(%o34)
```

$$-\frac{\sum_{n=1}^{\infty}\sin(\frac{\pi n x}{L})\left((2L^2\cos(\frac{\pi c n t}{L}) - 2L^2\cos(\omega t))\int_0^L r(x)\sin(\frac{\pi n x}{L})dx\right)}{\pi^2 L c^2 n^2 - L^3\omega^2} - \frac{\sum_{n=1}^{\infty}(2L^2\omega^2 - 2\pi^2 c^2 n^2)\sin(\frac{\pi n x}{L})\cos(\frac{\pi c n t}{L})\int_0^L f(x)\sin(\frac{\pi n x}{L})dx}{\pi^2 L c^2 n^2 - L^3\omega^2} \qquad (\%o34)$$

Since no assumptions were made about ω, the solution corresponds to the case without resonance ($\omega \neq \frac{cn\pi}{L}$).

Example 7.9 (⋆)**.** Here is another example with homogeneous boundary conditions and a driving term. This is Example 4.31 in [3].

$$\begin{cases} u_{tt} = c^2 u_{xx} + Ax \quad (x,t) \in [0,L] \times \mathbb{R}^+ \ A \in \mathbb{R} \\ u(0,t) = 0 \quad u(L,t) = 0 \\ u(x,0) = 0 \quad u_t(x,0) = 0 \end{cases}$$

Then:

```
(%i35) assume(L>0)$
(%i36) assume(t>0)$
(%i37) dirichlet_wave(a*x,0,0,0,0,x,t,L,c,inf);
(%o37)
```

$$\frac{\sum_{n=1}^{\infty} \frac{\left(2L^3 a\,(-1)^n \cos\left(\frac{\pi c n t}{L}\right) - 2L^3 a\,(-1)^n\right) \sin\left(\frac{\pi n x}{L}\right)}{n^3}}{\pi^3\, c^2}$$

This solution can be written in a simplified form:

$$\frac{AL^2x - Ax^3}{6c^2} + 2\sum_{n=1}^{\infty} \frac{(-1)^n L^3 A}{n^3 \pi^3 c^2} \sin\frac{n\pi x}{L} \cos\frac{n\pi t}{L}$$

since

```
(%i38) fouriersin_series((a*L^2*x-a*x^3)/6,x,L,inf);
```

$$\frac{2L^3 a \sum_{n=1}^{\infty} \frac{(-1)^{n+1} \sin\left(\frac{\pi n x}{L}\right)}{n^3}}{\pi^3}$$

Example 7.10 (⋆)**.** Let's do an example with mixed boundary conditions. The following is Exercise 12 of Section 4.2 in [3].

$$\begin{cases} u_{tt} = c^2 u_{xx} \quad (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(0,t) = 0 \quad u_x(L,t) = 0 \\ u(x,0) = f(x) \quad u_t(x,0) = 0 \end{cases}$$

The solution is given by:

```
(%i2) assume(c>0)$
(%i3) assume(L>0)$
(%i4) mixed_wave(0,f(x),0,1,0,0,1,0,0,x,t,L,c,inf);
(%o4)
```

$$\frac{2\sum_{n=1}^{\infty} \cos\left(\frac{\pi c\,(2n-1)t}{2L}\right) \sin\left(\frac{\pi(2n-1)x}{2L}\right) \int_0^L \mathrm{f}(x) \sin\left(\frac{\pi(2n-1)x}{2L}\right) dx}{L}$$

Bibliography

[1] N. Abbasi. *12000.org Solving Partial Differential Equations Using Maple 2019 and Mathematica 12*. 2019. URL: `https://www.12000.org/my_notes/pde_in_CAS/maple_2019_and_mma_12/index.htm` (visited on 04/02/2020).

[2] R. Haberman. *Applied Partial Differential Equations*. Pearson Education, 2013.

[3] J.D. Logan. *Applied Partial Differential Equations*. Springer, 2015.

[4] Maxima project. *Maxima Manual*. 2019. URL: `http://maxima.sourceforge.net/docs/manual/maxima.pdf`.

[5] J.Rubinstein Y. Pinchover. *An Introduction to Partial Differential Equations*. Cambridge, 2005.