SM472, Computational Fourier Transforms [1]

Class notes for a Capstone course taught Spring 2006-2007. The text for the course is the book *Fast Fourier transform* by James Walker [W1]. Examples using `SAGE` [S] illustrate the computations are given throughout.

Contents

Application of FS, FTs, and DFTs (including DCTs and FFTs), are found in

- the heat equation and the wave equation in physics,
- image compression,
- analog-to-digital conversion,
- spectroscopy (in medicine, for example),
- passive sonar (via frequency analysis),
- multiplication of very large integers (useful in cryptography),
- statistics,

among many others.

In fact, Patent number 5835392 is "Method for performing complex fast fourier transforms (FFT's)", and Patent number 6609140 is "Methods and apparatus for fast fourier transforms." These are just two of the many patents which refer to FFTs. Patent number 4025769 is "Apparatus for convoluting complex functions expressed as fourier series" and Patent number 5596661 is "Monolithic optical waveguide filters based on Fourier expansion." These are just two of the many patents which refer to FSs. You can look these up in Google's Patent search engine, `http://www.google.com/ptshp?hl=en&tab=wt&q=`

to read more details. You'll see related patents on the same page. Try "discrete cosine transform" or "digital filters" to see other devices which are related to the topic of this course.

1 Background on Fourier transforms

This section is not, strictly speaking, needed for the introduction of discrete Fourier transforms, it helps put things in the right conceptual framework.

1.1 Motivation

First, let's start with some motivation for the form of some of the definitions. Let's start with something you are very familiar with: power series $\sum_{t=0}^{\infty} f(t)x^t$, where we are writing the coefficients $f(t)$, $t \in \mathbb{N}$, of the power series in a functional notation instead of using subscripts. Usually, if $f(0)$, $f(1)$, $f(2)$, ... is a given sequence then the power series $\sum_{t=0}^{\infty} f(t)x^t$ is called the *generating function* of the sequence[2].

The continuous analog of a sum is an integral:

$$\sum_{t=0}^{\infty} f(t)x^t \rightsquigarrow \int_0^{\infty} f(t)x^t\,dt.$$

Replacing x by e^{-s}, the map

$$f \longmapsto \int_0^{\infty} f(t)x^t\,dt = \int_0^{\infty} f(t)e^{-st}\,dt,$$

is the *Laplace transform*. Replacing x by e^{-2iy}, the map

$$f \longmapsto \int_0^{\infty} f(t)x^t\,dt = \int_0^{\infty} f(t)e^{-2ity}\,dt,$$

is essentially the Fourier transform (the definition below includes a factor $\frac{1}{\sqrt{\pi}}$ for convenience). In other words, the Laplace transform and the Fourier transform are both continuous analogs of power series.

[2]In fact, since power series are so well understood, often times one studies the generating function to better understand a given sequence.

To go a little further with this analogy, suppose we have two power series $A(x) = \sum_{j=0}^{\infty} a(j)x^j$ and $B(x) = \sum_{k=0}^{\infty} B(k)x^k$. The product is given by

$$\begin{aligned} A(x)B(x) &= (\textstyle\sum_{j=0}^{\infty} a(j)x^j)(\sum_{k=0}^{\infty} B(k)x^k) \\ &= (a(0) + a(1)x + a(2)x^2 + ...)(b(0) + b(1)x + b(2)x^2 + ...) \\ &= a(0)b(0) + (a(0)b(1) + a(1)b(0))x + (a(2)b(0) + a(1)b(1) + a(0)b(2))x^2 + ... \\ &= \textstyle\sum_{m=0}^{\infty} c(m)x^m, \end{aligned}$$

where

$$c_m = \sum_{j+k=m} a(j)b(k) = \sum_{j=0}^{m} a(j)b(m-j).$$

This last expression is referred to as the *Dirichlet*[3] *convolution* of the a_j's and b_k's. Likewise, if $F(s) = \int_0^\infty f(t)e^{-st}\,dt$ and $G(s) = \int_0^\infty g(t)e^{-st}\,dt$ then

$$F(s)G(s) = \int_0^\infty (f * g)(t)e^{-st}\,dt,$$

where

$$(f * g)(t) = \int_0^t f(z)g(t-z)\,dz.$$

The above integral, $\int_0^\infty f(t)e^{-2ity}\,dt$, is over $0 < t < \infty$. Equivalently, it is an integral over $\mathbb{R}$, but restricted to functions which vanish off $(0, \infty)$. If you replace these functions supported on $(0, \infty)$ by any function, then we are lead to the transform

$$f \longmapsto \int_{-\infty}^{\infty} f(t)e^{-2ity}\,dt.$$

This is discussed in the next section.

1.2 The Fourier transform on $\mathbb{R}$

If f is any function on the real line $\mathbb{R}$ whose absolute value is integrable

$$||f||_1 = \int_{\mathbb{R}} |f(x)|\,dx < \infty$$

[3]Pronounced "Dear-ish-lay".

then we say f *is* L^1, written $f \in L^1(\mathbb{R})$. In this case, we define the *Fourier transform* of f by

$$F(f)(y) = \hat{f}(y) = \frac{1}{\sqrt{\pi}} \int_{\mathbb{R}} f(x) e^{-2ixy}\, dx.$$

This is a bounded function, written $f \in L^\infty(\mathbb{R})$,

$$||\hat{f}||_\infty = \sup_{y \in \mathbb{R}} |\hat{f}(y)| < \infty.$$

It can be shown that F (extends to a well-defined operator which) sends any square-integrable function to another square-integrable function. In other words,

$$F : L^2(\mathbb{R}) \to L^2(\mathbb{R}),$$

where $L^2(\mathbb{R})$ denotes the vector space (actually, a Hilbert space) of functions for which the L^2-norm is finite:

$$||f||_2 = (\int_{\mathbb{R}} |f(x)|^2\, dx)^{1/2} < \infty.$$

Example 1 Here is an example. Let $f_0(x) = e^{-x^2}$. We have

$$\begin{array}{ll} \hat{f}_0(y) & = \frac{1}{\sqrt{\pi}} \int_{\mathbb{R}} e^{-x^2} e^{-2ixy}\, dx \\ & = \frac{1}{\sqrt{\pi}} \int_{\mathbb{R}} e^{-y^2} e^{-(x+iy)^2}\, dx \\ & = \frac{1}{\sqrt{\pi}} e^{-y^2} \int_{Im(z)=y} e^{-z^2}\, dz \\ & = \frac{1}{\sqrt{\pi}} e^{-y^2} \int_{Im(z)=0} e^{-z^2}\, dz, \end{array}$$

where $z = x+iy$ the last equality follows from the Residue Theorem from complex analysis. Now, this is

$$\begin{array}{l} = \frac{1}{\sqrt{\pi}} e^{-y^2} \int_{Im(z)=0} e^{-z^2}\, dz \\ = \frac{1}{\sqrt{\pi}} e^{-y^2} \int_{\mathbb{R}} e^{-x^2}\, dx \\ = e^{-y^2}. \end{array}$$

This last line depends on

$$\int_{\mathbb{R}} e^{-x^2}\, dx = \sqrt{\pi}.$$

(Here's the quick proof of this:

$$
\begin{aligned}
(\textstyle\int_{\mathbb{R}} e^{-x^2}\,dx)^2 &= \textstyle\int_{\mathbb{R}}\int_{\mathbb{R}} e^{-x^2}e^{-y^2}\,dxdy \\
&= \textstyle\int_{\mathbb{R}}\int_{\mathbb{R}} e^{-x^2-y^2}\,dxdy \\
&= \textstyle\int_0^\infty \int_0^{2\pi} e^{-r^2} r\,d\theta dr \\
&= 2\pi \textstyle\int_0^\infty e^{-r^2} r\,dr \\
&= 2\pi\frac{1}{2} \textstyle\int_0^\infty e^{-u}\,du = \pi.
\end{aligned}
$$

Now take square-roots.)

This proves that this example satisfies

$$F(f_0)(y) = f_0(y). \tag{1}$$

In other words, that the operator F has eigenvector f_0 and eigenvalue $\lambda = 1$.

Exercises:

1. Verify that $F(f')(y) = 2iyF(f)(y)$, for all $f \in L^1(\mathbb{R}) \cap C^1(\mathbb{R})$. (Hint: Integrate by parts.)

2. If $f_1(x) = xe^{-x^2}$ then f_1 is an eigenvector of the Fourier transform with eigenvalue $\lambda = -i$, i.e., $F(f_1) = -if_1$. (Hint: Use (1) and the previous exercise.)

3. Let
$$f(x) = \begin{cases} 1, & |x| < 1/2, \\ 0, & |x| \geq 1/2, \end{cases}$$
and show that $\hat{f}(y) = \frac{\sin(\pi y)}{\pi y}$ (= sinc(y)).

4. For $a < 0$, let
$$f(x) = \begin{cases} e^{2\pi a x}, & x > 0, \\ 0, & x \leq 0, \end{cases}$$
and show that $\hat{f}(y) = \frac{1}{(2\pi)((iy-a)}$. Similarly, let
$$f(x) = \begin{cases} e^{-2\pi a x}, & x < 0, \\ 0, & x \geq 0, \end{cases}$$
and show that $\hat{f}(y) = \frac{-1}{(2\pi)((iy+a)}$.

5. For $a < 0$, let $f(x) = e^{2\pi a|x|}$ and show that $\hat{f}(y) = -\frac{a}{(\pi)((y^2+a^2)}$.

The following result spells out the main properties of the FT.

Theorem 2 *Let $f, g \in L^1(\mathbb{R})$, with $f' \in L^1(\mathbb{R})$ and $xf(x) \in L^1(\mathbb{R})$.*

- *Linearity:*

$$af + bg \stackrel{F}{\longmapsto} a\hat{f} + b\hat{g},$$

- *Scaling:*

$$f(x/a) \stackrel{F}{\longmapsto} a\hat{f}(ay),$$

and

$$f(ax) \stackrel{F}{\longmapsto} a^{-1}\hat{f}(y/a),$$

- *Shifting:*

$$f(x-a) \stackrel{F}{\longmapsto} e^{-2\pi i a y}\hat{f}(y),$$

- *Modulation:*

$$e^{2\pi i a x} f(x) \stackrel{F}{\longmapsto} \hat{f}(y-a),$$

- *Inversion:*

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(t) e^{2\pi i t x}\, dt.$$

- *Differentiation:*

$$f'(x) \stackrel{F}{\longmapsto} 2\pi i y \hat{f}(y),$$

and

$$xf(x) \stackrel{F}{\longmapsto} \frac{i}{2\pi}\hat{f}(y).$$

- *Parseval: For $f, g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, we have*

$$\int_{\mathbb{R}} f(x)\overline{g(x)}\, dx = \int_{\mathbb{R}} \hat{f}(y)\overline{\hat{g}(y)}\, dy,$$

and in particular,

$$\int_{\mathbb{R}} |f(x)|^2 \, dx = \int_{\mathbb{R}} |\hat{f}(y)|^2 \, dy.$$

This can be found in Chapter 5 of Walker [W1].

1.3 Application to Laplace's PDE

We shall use the FT to find a function $w(x,t)$ satisfying

$$\begin{cases} \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial t^2} = 0, \\ w(x,0) = f(x). \end{cases}$$

This represents the steady state temperature of a semi-infinite plate, in the shape of the upper half-plane, where the boundary (the x-axis) is kept heated to the temperature $f(x)$ at the point $(x,0)$ for all time.

solution: Assume that for each y, $w(x,y)$ is in $L^1(\mathbb{R})$, as a function of x. Let

$$\hat{w}(u,y) = \int_{\mathbb{R}} w(x,y) e^{-2\pi i x u} \, dx,$$

so

$$\hat{w}(u,0) = \int_{\mathbb{R}} w(x,0) e^{-2\pi i x u} \, dx, = \int_{\mathbb{R}} f(x) e^{-2\pi i x u} \, dx = \hat{f}(u),$$

by the initial condition, and

$$w(x,y) = \int_{\mathbb{R}} \hat{w}(u,y) e^{2\pi i x u} \, du, \tag{2}$$

by the inversion formula.

By (2), we have

$$w_{xx}(x,y) = \int_{\mathbb{R}} -(2\pi u)^2 \hat{w}(u,y) e^{2\pi i x u} \, du,$$

and

$$w_{yy}(x,y) = \int_{\mathbb{R}} \hat{w}_{yy}(u,y) e^{2\pi i x u} \, du,$$

so

$$0 = w_{xx}(x, y) + w_{yy}(x, y) = \int_{\mathbb{R}} (-(2\pi u)^2 \hat{w}(u, y) + \hat{w}_{yy}(u, y))e^{2\pi ixu}\, du.$$

Therefore, by Laplace's equation,

$$-(2\pi u)^2 \hat{w}(u, y) + \hat{w}_{yy}(u, y) = 0,$$

or

$$k''(y) - (2\pi u)^2 k(y) = 0,$$

where $k(y) = \hat{w}(u, y)$. This means

$$\hat{w}(u, y) = k(y) = c_1 e^{-2\pi|u|y} + c_2 e^{2\pi|u|y},$$

for some constants c_1, c_2. Let $c_2 = 0$ (because it works, that's why!) and solve for c_1 using the initial condition $\hat{w}(u, 0) = \hat{f}(u)$, to get

$$\hat{w}(u, y) = \hat{f}(u)e^{-2\pi|u|y}.$$

Using (2) again, we have

$$\begin{aligned} w(x, y) &= \int_{\mathbb{R}} \hat{f}(u)e^{-2\pi|u|y} e^{2\pi ixu}\, du \\ &= \int_{\mathbb{R}} (\int_{\mathbb{R}} f(z)e^{-2\pi izu}\, dz)e^{-2\pi|u|y} e^{2\pi ixu}\, du \\ &= \int_{\mathbb{R}} (\int_{\mathbb{R}} e^{-2\pi|u|y} e^{2\pi i(x-z)u}\, du) f(z)\, dz \\ &= \int_{\mathbb{R}} \frac{1}{\pi} \frac{y}{y^2+(x-z)^2} f(z)\, dz, \end{aligned}$$

which is the convolution of $f(x)$ with $K_y(x) = \frac{1}{\pi}\frac{y}{y^2+x^2}$. This function $w(x, y) = (f * K_y)(x)$ is the desired solution.

1.4 Application to Schödinger's PDE

Assume $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$.

We shall use the FT to find a function $\psi(x, t)$ satisfying

$$\begin{cases} a^2 \frac{\partial^2 \psi}{\partial x^2} = \frac{\partial \psi}{\partial t}, \\ \psi(x, 0) = f(x), \end{cases}$$

where (for convenience) $a^2 = \frac{ih}{2m}$, and h is Planck's constant. The function $\psi(x,t)|^2$, where ψ is the solution (or "wave function") of the above PDE, represents (under certain conditions on f) the probability density function of a free particle on a semi-infinite plate, in the shape of the upper half-plane, where the distribution on boundary (the x-axis) is determined by $f(x)$.

solution: Assume that for each t, $\psi(x,t)$ is in $L^1(\mathbb{R})$, as a function of x. Let

$$\hat{\psi}(u,t) = \int_{\mathbb{R}} \psi(x,t)e^{-2\pi ixu}\,dx,$$

so

$$\hat{\psi}(u,0) = \int_{\mathbb{R}} \psi(x,0)e^{-2\pi ixu}\,dx, = \int_{\mathbb{R}} f(x)e^{-2\pi ixu}\,dx = \hat{f}(u),$$

by the initial condition, and

$$\psi(x,t) = \int_{\mathbb{R}} \hat{\psi}(u,t)e^{2\pi ixu}\,du, \tag{3}$$

by the inversion formula. By (3), we have

$$\psi_{xx}(x,t) = \int_{\mathbb{R}} -(2\pi u)^2\hat{\psi}(u,t)e^{2\pi ixu}\,du,$$

and

$$\psi_t(x,t) = \int_{\mathbb{R}} \hat{\psi}_t(u,t)e^{2\pi ixu}\,du,$$

so

$$0 = a^2\psi_{xx}(x,t) - \psi_t(x,t) = \int_{\mathbb{R}} (-a^2(2\pi u)^2\hat{\psi}(u,t) - \hat{\psi}_t(u,t))e^{2\pi ixu}\,du.$$

By Schödinger's PDE,

$$-a^2(2\pi u)^2\hat{\psi}(u,t) - \hat{\psi}_t(u,t) = 0,$$

or

$$k'(t) = -a^2(2\pi u)^2 k(t),$$

where $k(t) = \hat{\psi}(u,t)$. This means

$$\hat{\psi}(u,t) = k(t) = c_0 e^{-a^2(2\pi u)^2 t},$$

for some constant c_0. Using the above initial condition to sove for c_0, we obtain $\hat{\psi}(u,t) = \hat{f}(u)e^{-a^2(2\pi u)^2 t}$. Using (3) again, we have

$$\begin{array}{ll} \psi(x,t) & = \int_{\mathbb{R}} \hat{f}(u)e^{-a^2(2\pi u)^2 t} e^{2\pi i x u}\, du \\ & = \int_{\mathbb{R}} (\int_{\mathbb{R}} f(z)e^{-2\pi i z u}\, dz) e^{-a^2(2\pi u)^2 t} e^{2\pi i x u}\, du \\ & = \int_{\mathbb{R}} (\int_{\mathbb{R}} e^{-a^2(2\pi u)^2 t} e^{2\pi i (x-z) u}\, du) f(z)\, dz \\ & = \int_{\mathbb{R}} H_t(x-z) f(z)\, dz, \end{array}$$

which is the convolution of $f(x)$ with $H_t(x) = \int_{\mathbb{R}} e^{-a^2(2\pi u)^2 t} e^{2\pi i x u}\, du$. This function $w(x,y) = (f * H_t)(x)$ is the desired solution.

By the Plancheral formula,

$$\begin{array}{ll} \int_{\mathbb{R}} |\psi(x,t)|^2\, dx & = \int_{\mathbb{R}} |\hat{\psi}(u,t)|^2\, du \\ & = \int_{\mathbb{R}} |\hat{f}(u)e^{-a^2(2\pi u)^2 t}|^2\, du \\ & = \int_{\mathbb{R}} |\hat{f}(u)|^2\, du \\ & = \int_{\mathbb{R}} |f(x)|^2\, dx \\ & = \int_{\mathbb{R}} |\psi(u,0)|^2\, dx, \end{array}$$

by the initial condition. In particular,if f has L^2-norm equal to 1 then so does $\psi(x,t)$, for each $t \geq 0$.

2 Fourier series

History: Fourier series were discovered by J. Fourier, a Frenchman who was a mathematician among other things. In fact, Fourier was Napolean's scientific advisor during France's invasion of Egypt in the late 1800's. When Napolean returned to France, he "elected" (i.e., appointed) Fourier to be a Prefect - basically an important administrative post where he oversaw some large construction projects, such as highway constructions. It was during this time when Fourier worked on the theory of heat on the side. His solution to the heat equation is basically what we teach in the last few weeks of your differential equations course. The exception being that our understanding of Fourier series now is much better than what was known in the early 1800's. For example, Fourier did not know of the integral formulas (4), (5) for the Fourier coefficients given below.

Motivation: Fourier series, since series, and cosine series are all expansions for a function $f(x)$ in terms of trigonometric functions, much in the same way that a Taylor series is an expansion in terms of power functions. Both Fourier and Taylor series can be used to approximate a sufficient;y "nice" function $f(x)$. There are at least three important differences between the two types of series. (1) For a function to have a Taylor series it must be differentiable[4], whereas for a Fourier series it does not even have to be continuous. (2) Another difference is that the Taylor series is typically not periodic (though it can be in some cases), whereas a Fourier series is *always* periodic. (3) Finally, the Taylor series (when it converges) always converges to the function $f(x)$, but the Fourier series may not (see Dirichlet's theorem below for a more precise description of what happens).

Given a "nice" periodic function $f(x)$ of period P, there are a_n with $n \geq 0$ and b_n with $n \geq 1$ such that $f(x)$ has ("real") Fourier series

$$FS_{\mathbb{R}}(f)(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(\frac{2\pi nx}{P}) + b_n \sin(\frac{2\pi nx}{P})].$$

These *Fourier series coefficients* are given by

$$a_n = \frac{2}{P} \int_0^P f(x) \cos(\frac{2\pi nx}{P})\, dx, \tag{4}$$

and

$$b_n = \frac{2}{P} \int_0^P f(x) \sin(\frac{2\pi nx}{P})\, dx. \tag{5}$$

What does "nice" mean? When does the series converge? When it does converge, what is the relationship between f and $FS_{\mathbb{R}}(f)$?

Definition 3 A function $f(x)$ on a finite interval $[a, b]$ is said to be *of bounded variation* if there is a constanct $C > 0$ such that for any partition $x_0 = a < x_1 < \ldots < x_{n-1} < x_n = b$ of the interval (a, b), we have

$$\sum_{i=0}^{n-1} |f(x_{i+1}) - f(x_i)| \leq C.$$

[4]Remember the formula for the n-th Taylor series coefficient centered at $x = a$ - $a_n = \frac{f^{(n)}(a)}{n!}$?

Another characterization states that the functions of bounded variation on a closed interval are exactly those functions which can be written as a difference $g - h$, where both g and h are monotone.

Let $BV([0, P])$ denote the vector space of functions on $[0, P]$ of bounded variation.

Since any monotone function is Riemann integrable, so is any function of bounded variation. The map FS is therefore well-defined on elements of $BV([0, P])$.

Likewise, given a "nice" periodic function $f(x)$ of period P, there are a_n with $n \geq 0$ and b_n with $n \geq 1$ such that $f(x)$ has ("complex") Fourier series

$$FS(f)(x) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{2\pi inx}{P}}.$$

These *Fourier series coefficients* are given by

$$c_n = \frac{1}{P} \int_0^P f(x) e^{-2\pi inx/P}\, dx, \tag{6}$$

for $n \in \mathbb{Z}$.

Theorem 4 *If $f \in BV([0, P])$ and if f has a Fourier series representation*

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{2\pi inx}{P}}$$

which converges absolutely then (6) holds.

proof: By hypothesis,

$$\int_0^P f(x) e^{-2\pi inx/P}\, dx = \int_0^P \sum_{m=-\infty}^{\infty} c_m e^{\frac{2\pi imx}{P}} e^{-2\pi inx/P}\, dx.$$

Interchanging the sum and the integral, this is

$$= \sum_{m=-\infty}^{\infty} c_m \int_0^P e^{2\pi i(m-n)x/P}\, dx = Pc_n.$$

$\square$

In the above proof, we have used the fact that $\{e^{2\pi inx/P}\}_{n \in \mathbb{Z}}$ forms a sequence of periodic orthogonal functions on the interval $(0, P)$.

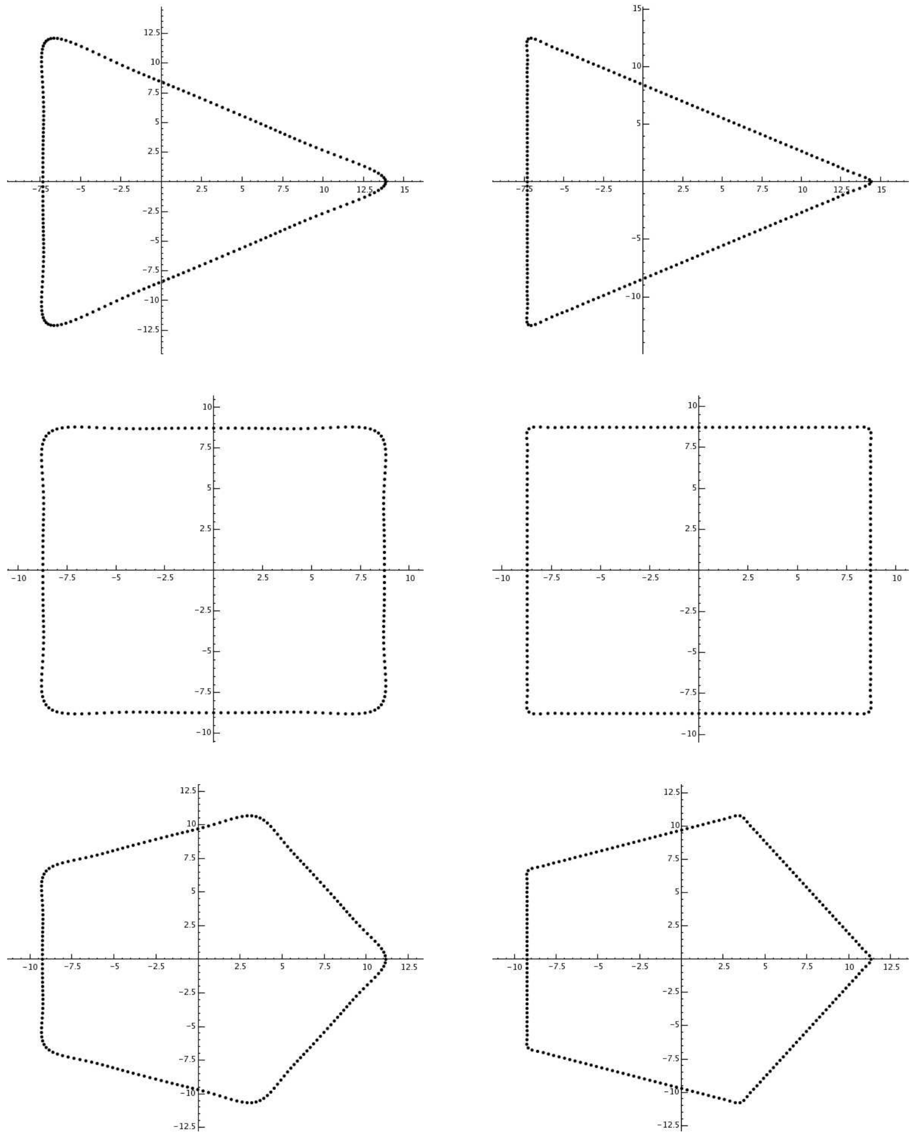
Example 5 Given any complex Fourier series $f(t) = x(t) + iy(t)$, you can plot the parametric curve $(x(t), y(t)$, for $0 < t < P$. A. Robert [R] classifies the complex Fourier series whose parametric plot is a regular polygon. The n-gons are all obtained by transforming in various ways (such as translations and reflections) the basic Fourier series

$$f(t) = \sum_{k\in\mathbb{Z}} (1 + kn)^{-2} e^{\pi i(1+kn)t}.$$

Here are some examples. In the first row of the table of plots below, we plot the partial sum

$$\sum_{|k|<N} (1 + kn)^{-2} e^{\pi i(1+kn)t}$$

with $N = 3$, $n = 3$, then $N = 10$, $n = 3$. The next line is with $N = 3$, $n = 4$, then $N = 10$, $n = 4$; the last line is with $N = 3$, $n = 5$, then $N = 10$, $n = 5$.



Here is the SAGE code which produced these examples:

```
sage: N = 3
sage: L = [1+5*k for k in range(-N,N)]
sage: f = lambda t: sum([10*n^(-2)*exp(n*pi*I*t) for n in L])
sage: pts = [[real(f(i/100)),imag(f(i/100))] for i in range(200)]
sage: show(list_plot(pts))
sage:
```

```
sage: N = 10
sage: L = [1+5*k for k in range(-N,N)]
sage: f = lambda t: sum([10*n^(-2)*exp(n*pi*I*t) for n in L])
sage: pts = [[real(f(i/100)),imag(f(i/100))] for i in range(200)]
sage: show(list_plot(pts))
```

This code produces the pentagon pictures above. Changing the terms in the partial FS obviously changes the graph. For instance, replacing `L = [1+5*k for k in range(-N,N)]` by `L = [2+5*k+k**2 for k in range(-N,N)]` produces a figure looking like a goldfish!

The relationship between the real and the complex Fourier series is as follows: In $FS(f)$, replace the cosine and sine terms by

$$\cos(\frac{2\pi nx}{P}) = \frac{e^{2\pi inx/P}+e^{-2\pi inx/P}}{2},$$

and

$$\sin(\frac{2\pi nx}{P}) = \frac{e^{2\pi inx/P}-e^{-2\pi inx/P}}{2i}.$$

This implies $FS_{\mathbb{R}}(f) = FS(f)$, where

$$c_n = \begin{cases} \frac{a_0}{2}, & n=0, \\ \frac{a_n}{2}+\frac{b_n}{2i}, & n>0, \\ \frac{a_{-n}}{2}-\frac{b_{-n}}{2i}, & n<0. \end{cases}$$

In particular, $FS_{\mathbb{R}}(f)(x)$ converges if and only if $FS(f)(x)$ does.

Let $C^m(P)$ denote the vector space of all functions which are m-times continuously differentiable and have period P.

Lemma 6 *If $f \in C^m(P)$ then there is a constant $A = A(f) > 0$ such that $|c_n| \leq A_f|n|^{-m}$, for all $n \neq 0$.*

proof: Integrate-by-parts:

$$\begin{aligned} \int_0^P f(x)e^{-2\pi inx/P}\,dx &= \left(\tfrac{P}{-2\pi in}\right) f(x)e^{-2\pi inx/P}|_{x=0}^{x=P} - \left(\tfrac{P}{-2\pi in}\right)\int_0^P f'(x)e^{-2\pi inx/P}\,dx \\ &= \left(\tfrac{P}{-2\pi in}\right)\int_0^P f'(x)e^{-2\pi inx/P}\,dx, \end{aligned}$$

since $f(0) = f(P)$.

Now use the "trivial" estimate

$$|\int_0^P g(x)e^{-2\pi inx/P}\,dx| \leq P \cdot \max_{0\leq x\leq P} |g(x)|$$

to get

$$|c_n| \leq \frac{\max_{0\leq x\leq P} |f'(x)|}{2\pi}\frac{1}{n}.$$

The proof is completed using mathematical induction (or simply repeat this process $m-1$ more times). $\Box$

Let $\ell^\infty(\mathbb{Z})$ denote the space of vectors $(x_n)_{n\in\mathbb{Z}}$ whose coordinates x_n are bounded and let

$$||(x_n)_{n\in\mathbb{Z}}|| = ||(x_n)_{n\in\mathbb{Z}}||_\infty = \max_{n\in\mathbb{Z}} |x_n|.$$

In particular, it follows that the Fourier transform defines a linear mapping

$$FS : C^0(P) \to \ell^\infty(\mathbb{Z}),$$

given by sending a continuous periodic function $f \in C^0(P)$ to its sequence of Fourier series coefficients $(c_n(f))_{n\in\mathbb{Z}} \in \ell^\infty(\mathbb{Z})$.

Theorem 7 *(Jordan) If $f \in BV([0, P])$ then $FS(f)(x)$ converges to $\frac{f(x+)+f(x-)}{2}$.*

proof: See §13.232 in [T].$\Box$

Perhaps more familiar is the following result, which was covered in your Differential Equations course.

Theorem 8 *(Dirichlet) If f has a finite number of maxima, a finite number of minima, and a finite number of discontinuities on the interval $(0, P)$ then $FS(f)(x)$ converges to $\frac{f(x+)+f(x-)}{2}$.*

proof: See §13.232 in [T].$\Box$

Example 9 *Let $P = 2$ and let $f(x) = x$ for $0 < x \leq 2$. This is sometimes referred to as the "sawtooth function", due to the shape of its graph. The Fourier coefficients are given by*

$$c_n = \frac{1}{2}\int_0^2 xe^{-\pi inx}\,dx = \frac{1}{2}[(\frac{1}{-\pi in})xe^{-\pi inx} - \frac{1}{(-\pi in)^2}e^{-\pi inx}]\Big|_{x=0}^{x=2} = \frac{1}{-\pi in},$$

when $n \neq 0$, and

$$c_0 = \frac{1}{2}\int_0^2 x\,dx = \frac{x^2}{4}\Big|_{x=0}^{x=2} = 1.$$

The Fourier series

$$FS(f)(x) = \sum_{n\in\mathbb{Z}} c_n e^{\pi inx} = 1 - \frac{1}{\pi i}\sum_{n\in\mathbb{Z},\ n\neq 0} \frac{1}{n}e^{\pi inx}$$

does not converge absolutely.

Example 10 *Let $P = 1$ and let $f(x) = 10x^2(1-x)^2$ for $0 < x \leq 1$. This function belongs to $C^1(1)$.*

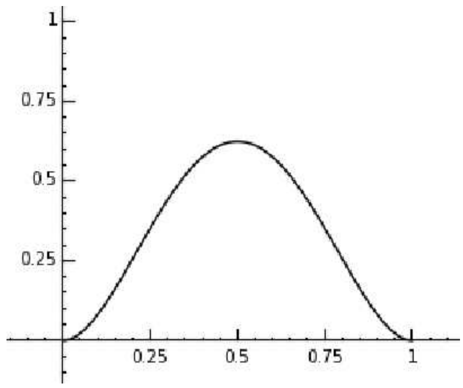


Figure 1: Graph of $f(x)$, $0 < x < 1$.

The Fourier coefficients are given by

$$c_n = \int_0^1 10x^2(1-x)^2 e^{-2\pi inx}\,dx = 10(\frac{-n^2\pi^2 - 3\,i\,n\pi + 3}{4\,i\,n^5\pi^5} - \frac{(-n^2\pi^2 + 3\,i\,n\pi + 3)}{4\,i\,n^5\pi^5}),$$

so $c_n = -\frac{30}{2\pi^4}n^{-4}$ when $n \neq 0$. When $n = 0$, we have

$$c_0 = \int_0^1 10x^2(1-x)^2\,dx = 1/3.$$

In this case, the Fourier series

$$FS(f)(x) = \sum_{n\in\mathbb{Z}} c_n e^{\pi i n x} = \frac{1}{3} - \frac{30}{2\pi^4} \sum_{n\in\mathbb{Z},\ n\neq 0} \frac{1}{n^4} e^{2\pi i n x}$$

does converge absolutely. In fact, it can even be differentiated term-by-term.

The `SAGE` *commands*

```
sage: f = maxima("10*x**2*(1-x)**2*exp(-2*Pi*i*x*n)")
sage: f.integral('x', 0, 1)
10*((i^2*n^2*Pi^2 - 3*i*n*Pi + 3)/(4*i^5*n^5*Pi^5) - (i^2*n^2*Pi^2 + 3*i*n*Pi + 3)*%e^-(2*i*n*Pi)/(4*i^5*n^5*Pi^5))
```

were used to compute the Fourier coefficients above.

2.1 Sine series and cosine series

Recall, to have a Fourier series you must be given two things: (1) a period $P = 2L$, (2) a function $f(x)$ defined on an interval of length $2L$, usually we take $-L < x < L$ (but sometimes $0 < x < 2L$ is used instead). The Fourier series of $f(x)$ with period $2L$ is

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

where a_n and b_n are as in (4), (5).

First, we discuss cosine series. To have a cosine series you must be given two things: (1) a period $P = 2L$, (2) a function $f(x)$ defined on the interval of length L, $0 < x < L$. The **cosine series of** $f(x)$ **with period** $2L$ is

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\frac{n\pi x}{L}),$$

where a_n is given by

$$a_n = \frac{2}{L} \int_0^L \cos(\frac{n\pi x}{L}) f(x)\, dx.$$

The cosine series of $f(x)$ is exactly the same as the Fourier series of the *even extension* of $f(x)$, defined by

$$f_{even}(x) = \begin{cases} f(x), & 0 < x < L, \\ f(-x), & -L < x < 0. \end{cases}$$

Next, we define sine series. To have a sine series you must be given two things: (1) a "period" $P = 2L$, (2) a function $f(x)$ defined on the interval of length L, $0 < x < L$. The *sine series of* $f(x)$ *with period* $2L$ is

$$f(x) \sim \sum_{n=1}^{\infty} b_n \sin(\frac{n\pi x}{L}),$$

where b_n is given by

$$b_n = \frac{2}{L} \int_0^L \sin(\frac{n\pi x}{L}) f(x)\ dx.$$

The sine series of $f(x)$ is exactly the same as the Fourier series of the *odd extension* of $f(x)$, defined by

$$f_{odd}(x) = \begin{cases} f(x), & 0 < x < L, \\ -f(-x), & -L < x < 0. \end{cases}$$

One last definition: the symbol $\sim$ is used above instead of $=$ because of the fact that was pointed out above: the Fourier series may not converge to $f(x)$ at every point (recall Dirichlet's Theorem 8).

Example 11 If $f(x) = 2 + x$, $-2 < x < 2$, is extended periodically to $\mathbb{R}$ with period 4 then $L = 2$. Without even computing the Fourier series, we can evaluate the FS using Dirichlet's theorem (Theorem 8 above).

Question: Using periodicity and Dirichlet's theorem, find the value that the Fourier series of $f(x)$ converges to at $x = 1, 2, 3$. (Ans: $f(x)$ is continuous at 1, so the FS at $x = 1$ converges to $f(1) = 3$ by Dirichlet's theorem. $f(x)$ is not defined at 2. It's FS is periodic with period 4, so at $x = 2$ the FS converges to $\frac{f(2+)+f(2-)}{2} = \frac{0+4}{2} = 2$. $f(x)$ is not defined at 3. It's FS is periodic with period 4, so at $x = 3$ the FS converges to $\frac{f(-1)+f(-1+)}{2} = \frac{1+1}{2} = 1$.)

The formulas for a_n and b_n enable us to compute the Fourier series coefficients a_0, a_n and b_n. These formulas give that the Fourier series of $f(x)$ is

$$f(x) \sim 4 + \sum_{n=1}^{\infty} -4\, \frac{n\pi\ \cos(n\pi)}{n^2\pi^2} \sin(\frac{n\pi x}{2}).$$

The Fourier series approximations to $f(x)$ are

$$S_0 = 2,\ S_1 = 2 + \frac{4}{\pi}\sin(\frac{\pi x}{2}),\ S_2 = 2 + 4\,\frac{\sin\left(\frac{1}{2}\,\pi\, x\right)}{\pi} - 2\,\frac{\sin(\pi\, x)}{\pi},\ \ldots$$

The graphs of each of these functions get closer and closer to the graph of $f(x)$ on the interval $-2 < x < 2$. For instance, the graph of $f(x)$ and of S_8 are given below:

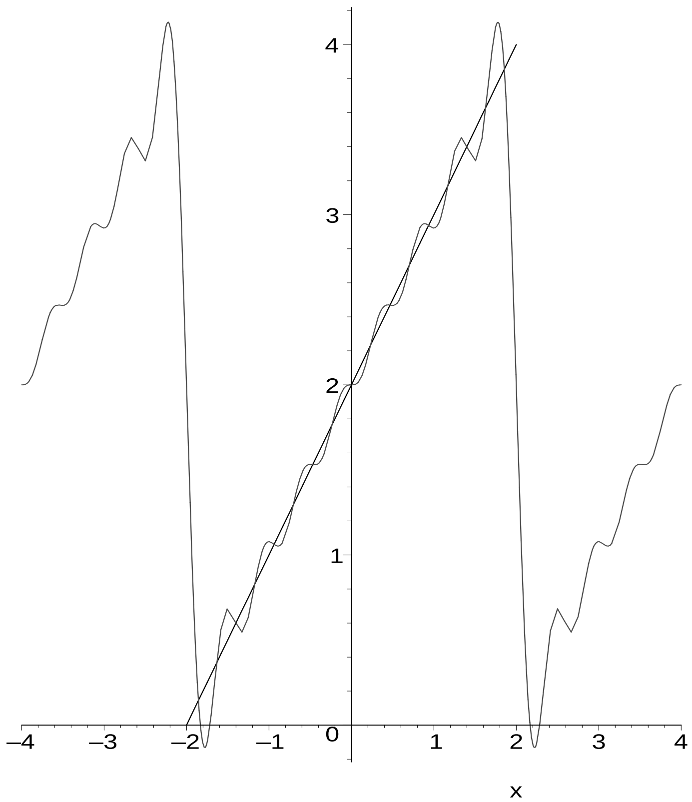


Figure 2: Graph of $f(x)$ and a Fourier series partial sum approximation of $f(x)$.

Notice that $f(x)$ is only defined from $-2 < x < 2$ yet the Fourier series is not only defined everywhere but is periodic with period $P = 2L = 4$. Also, notice that S_8 is not a bad approximation to $f(x)$, especially away from its jump discontinuities.

Example 12 This time, let's consider an example of a cosine series. In this case, we take the piecewise constant function $f(x)$ defined on $0 < x < 3$ by

$$f(x) = \begin{cases} 1, & 0 < x < 2, \\ -1, & 2 \leq x < 3. \end{cases}$$

We see therefore $L = 3$. The formula above for the cosine series coefficients gives that

$$f(x) \sim \frac{1}{3} + \sum_{n=1}^{\infty} 4\,\frac{\sin\left(\frac{2}{3}\,n\pi\right)}{n\pi}\cos(\frac{n\pi x}{3}).$$

The first few partial sums are

$$S_2 = 1/3 + 2\,\frac{\sqrt{3}\cos\left(\frac{1}{3}\,\pi\,x\right)}{\pi},$$

$$S_3 = 1/3 + 2\,\frac{\sqrt{3}\cos\left(\frac{1}{3}\,\pi\,x\right)}{\pi} - \frac{\sqrt{3}\cos\left(\frac{2}{3}\,\pi\,x\right)}{\pi}, \ldots$$

Also, notice that the cosine series approximation S_{10} is an even function but $f(x)$ is not (it's only defined from $0 < x < 3$). As before, the more terms in the cosine series we take, the better the approximation is, for $0 < x < 3$. For instance, the graph of $f(x)$ and of S_{10} are given below:

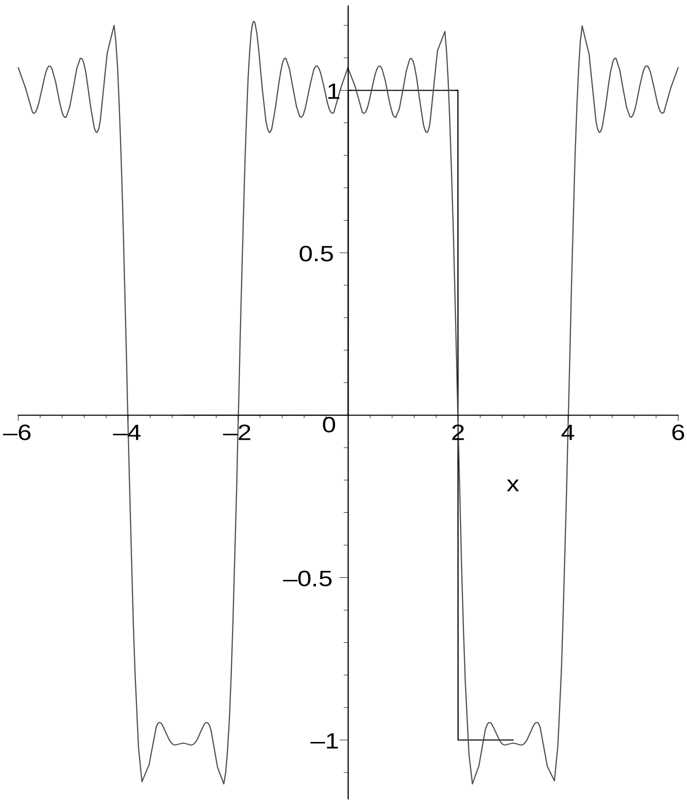


Figure 3: Graph of $f(x)$ and a cosine series approximation of $f(x)$.

Roughly speaking, the more (everywhere) differentiable the function is, the faster the Fourier series converges and, therefore, the better the partial sums of the Fourier series will approximate $f(x)$.

Example 13 Finally, let's consider an example of a sine series. In this case, we take the piecewise constant function $f(x)$ defined on $0 < x < 3$ by the same expression we used in the cosine series example above.

Question: Using periodicity and Dirichlet's theorem, find the value that the sine series of $f(x)$ converges to at $x = 1, 2, 3$. (Ans: $f(x)$ is continuous at 1, so the FS at $x = 1$ converges to $f(1) = 1$. $f(x)$ is not continuous at 2, so at $x = 2$ the SS converges to $\frac{f(2+)+f(2-)}{2} = \frac{f(-2+)+f(2-)}{2} = \frac{-1+1}{2} = 0$. $f(x)$ is not defined at 3. It's SS is periodic with period 6, so at $x = 3$ the SS converges to $\frac{f_{odd}(3-)+f_{odd}(3+)}{2} = \frac{-1+1}{2} = 0$.)

The formula above for the sine series coefficients give that

$$f(x) \sim \sum_{n=1}^{\infty} 2\,\frac{\cos(n\pi) - 2\cos\left(\frac{2}{3}\,n\pi\right) + 1}{n\pi}\,\sin\left(\frac{n\pi x}{3}\right).$$

The partial sums are

$$S_2 = 2\,\frac{\sin\left(1/3\,\pi\,x\right)}{\pi} + 3\,\frac{\sin\left(\frac{2}{3}\,\pi\,x\right)}{\pi},$$

$$S_3 = 2\,\frac{\sin\left(\frac{1}{3}\,\pi\,x\right)}{\pi} + 3\,\frac{\sin\left(\frac{2}{3}\,\pi\,x\right)}{\pi} - 4/3\,\frac{\sin\left(\pi\,x\right)}{\pi}, \ldots$$

These partial sums S_n, as $n \to \infty$, converge to their limit about as fast as those in the previous example. Instead of taking only 10 terms, this time we take 40. Observe from the graph below that the value of the sine series at $x = 2$ does seem to be approaching 0, as Dirichlet's Theorem predicts. The graph of $f(x)$ with S_{40} is

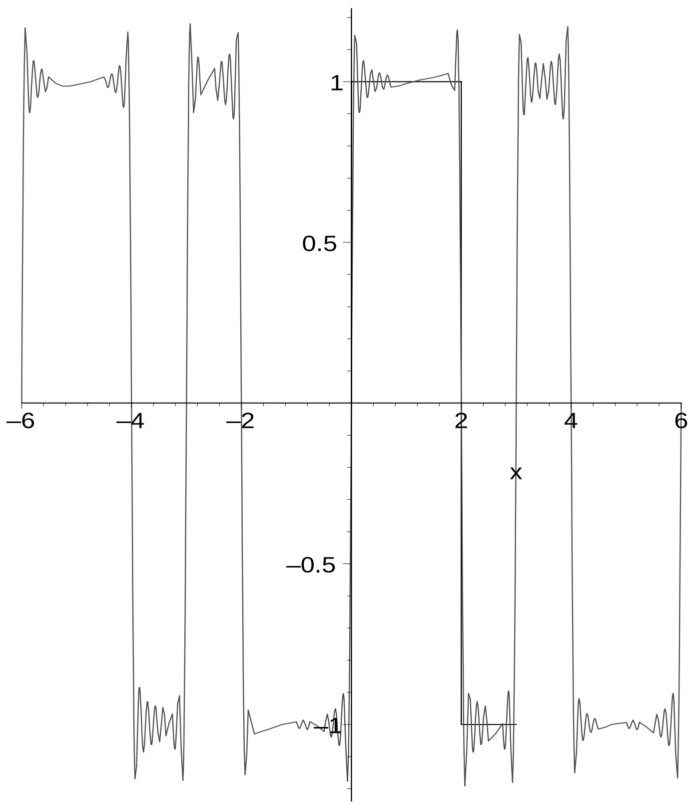


Figure 4: Graph of $f(x)$ and a sine series approximation of $f(x)$.

2.2 Exercises in Fourier series using SAGE

1. Let

$$f(x) = \begin{cases} 2, & 3/2 \le x \le 3, \\ 0, & 0 \le x \le 3/2, \\ -1, & -3 < x < 0, \end{cases}$$

period 6.

- Find the Fourier series of $f(x)$ and graph the function the FS converges to, $-3 \le x \le 3$.
- Write down the series in summation notation and compute the first 3 non-zero terms.

solution: Of course the Fourier series of $f(x)$ with period $2L$ is

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

where a_n and b_n are

$$a_n = \frac{1}{L}\int_{-L}^{L} \cos(\frac{n\pi x}{L}) f(x)\, dx,$$

$$b_n = \frac{1}{L}\int_{-L}^{L} \sin(\frac{n\pi x}{L}) f(x)\, dx.$$

Generally, the following functions compute the Fourier series coefficients of $f(x)$:

```
def fourier_series_cos_coeff(fcn,n,L):  # n>= 0
    lowerlimit = -L
    upperlimit = L
    an = maxima('tldefint(%s*cos(%s*n*x/%s),x,%s,%s)/L'%(fcn,maxima(pi),L,lowerlimit,upperlimit))
    return str(an).replace("%","")

def fourier_series_sin_coeff(fcn,n,L):  # n > 0
    lowerlimit = -L
    upperlimit = L
    bn = maxima('tldefint(%s*cos(%s*n*x/%s),x,-%s,%s)/L'%(fcn,maxima(pi),L,lowerlimit,upperlimit))
    return str(bn).replace("%","")
```

However, Maxima's support for piecewise defined functions is rudimentary and the above functions will not give us what we want. So we compute them "by hand" but with some help from SAGE [S] (and Maxima, whichis included with SAGE):

```
## n > 0 FS cosine coeff
fcn = 2; lowerlimit = 3/2; upperlimit = 3; L = 3
an1 = maxima('tldefint(%s*cos(%s*n*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an1
# 6*sin(%pi*n)/(%pi*n) - 6*sin(%pi*n/2)/(%pi*n)
fcn = -1; lowerlimit = -3; upperlimit = 0; L = 3
an2 = maxima('tldefint(%s*cos(%s*n*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an2
# -3*sin(%pi*n)/(%pi*n)
an = (an1+an2)/L
# (3*sin(%pi*n)/(%pi*n) - 6*sin(%pi*n/2)/(%pi*n))/3
```

In other words, for $n > 0$, $a_n = 2\frac{\sin(\pi n/2)}{\pi n}$.

```
## n = 0 FS cosine coeff
fcn = 2; lowerlimit = 3/2; upperlimit = 3; L = 3
an1 = maxima('tldefint(%s*cos(%s*0*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an1
# 3
fcn = -1; lowerlimit = -3; upperlimit = 0; L = 3
an2 = maxima('tldefint(%s*cos(%s*0*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an2
# -3
an = (an1+an2)/L
# 0
```

In other words, $a_0 = 0$.

```
## n > 0 FS sine coeff
fcn = 2; lowerlimit = 3/2; upperlimit = 3; L = 3
bn1 = maxima('tldefint(%s*sin(%s*n*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); bn1
# 6*cos(%pi*n/2)/(%pi*n) - 6*cos(%pi*n)/(%pi*n)
fcn = -1; lowerlimit = -3; upperlimit = 0; L = 3
bn2 = maxima('tldefint(%s*sin(%s*n*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); bn2
# 3/(%pi*n) - 3*cos(%pi*n)/(%pi*n)
bn = (bn1+bn2)/L
# ( - 9*cos(%pi*n)/(%pi*n) + 6*cos(%pi*n/2)/(%pi*n) + 3/(%pi*n))/3
```

In other words, $b_n = \frac{1-3(-1)^n+2\cos(\pi n/2)}{\pi n}$.

The Fourier series is therefore

$$f(x) \sim \sum_{n=1}^{\infty}[(2\frac{\sin(\pi n/2)}{\pi n})\cos(\frac{n\pi x}{3})+(\frac{1-3(-1)^n+2\cos(\pi n/2)}{\pi n})\sin(\frac{n\pi x}{3})],$$

```
sage: Pi = RR(pi)
sage: bn = lambda n:( - 9*cos(Pi*n)/(Pi*n) + 6*cos(Pi*n/2)/(Pi*n) + 3/(Pi*n))/3
sage: bn(1); bn(2); bn(3)
 1.2732395447351628
 -0.63661977236758127
 0.42441318157838753

sage: an = lambda n:2*(sin(Pi*n/2))/(Pi*n)
sage: an(1); an(2); an(3)
 0.63661977236758138
 0.000000000000000038981718325193755
 -0.21220659078919379
```

Here are the first few numerical values of these coefficients:

$$
\begin{aligned}
&a_1 = \tfrac{2}{\pi} = 0.6366197723675813...,\\
&a_2 = 0,\\
&a_3 = -\tfrac{2}{3\pi} = -0.2122065907891937...,\\
&b_1 = \tfrac{4}{\pi} = 1.273239544735162...,\\
&b_2 = \tfrac{-2}{\pi} = -0.6366197723675812...,\\
&b_3 = \tfrac{4}{\pi 3} = 0.4244131815783875... \,.
\end{aligned}
$$

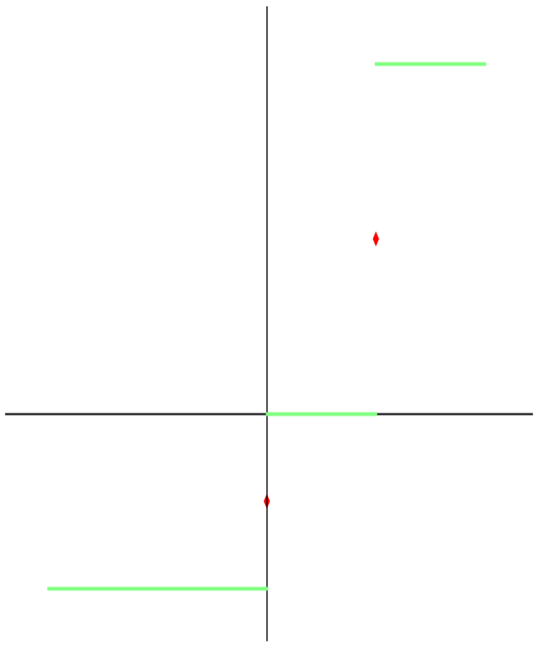


Figure 5: Graph of Fourier series of $f(x)$.

2. Let

$$f(x) = \begin{cases} 2, & 3/2 \le t \le 3, \\ 0, & 0 \le t \le 3/2, \end{cases}$$

- Find the Cosine series of $f(x)$ (period 6) and graph the function the CS converges to, $-3 \le x \le 3$.
- Write down the series in summation notation and compute the first 3 non-zero terms.

solution: Of course the Cosine series of $f(x)$ with period $2L$ is

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\frac{n\pi x}{L}),$$

where a_n is

$$a_n = \frac{2}{L}\int_0^L \cos(\frac{n\pi x}{L}) f(x)\ dx.$$

A simple Python program:

```
def cosine_series_coeff(fcn,n,L):   # n>= 0
    lowerlimit = 0
    upperlimit = L
    an = maxima('2*tldefint(%s*cos(%s*n*x/%s),x,%s,%s)/L'%(fcn,maxima(pi),L,lowerlimit,upperlimit))
    return str(an).replace("%","")
```

It was noted above that this program will not help us, for the type of function we are dealing with here. So, we have SAGE do the computations "piece-by-piece":

```
## n > 0
fcn = 2; lowerlimit = 3/2; upperlimit = 3; L = 3
an1 = maxima('tldefint(%s*cos(%s*n*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an1
# 6*sin(%pi*n)/(%pi*n) - 6*sin(%pi*n/2)/(%pi*n)
an = (2/L)*an1
an
# 2*(6*sin(%pi*n)/(%pi*n) - 6*sin(%pi*n/2)/(%pi*n))/3
```

In other words, $a_n = 4\frac{\sin(\pi n/2)}{\pi n}$. To find the 0-th coefficient, use the commands

```
## n = 0
an1 = maxima('tldefint(%s*cos(%s*0*x/%s),x,%s,%s)'%(fcn,maxima(pi),L,lowerlimit,upperlimit)); an1
# 3
an = (2/L)*an1
an
# 2
# a0 = 2
```

or the command

```
sage: f1 = lambda x:2
sage: f2 = lambda x:0
sage: f = Piecewise([[(0,3/2),f2],[(3/2,3),f1]])
sage: f.cosine_series_coefficient(0,3)
      2
```

In other words, $a_0 = 2$.

The Cosine series is therefore

$$f(x) \sim 1 + \sum_{n=1}^{\infty}(4\frac{\sin(\pi n/2)}{\pi n})\cos(\frac{n\pi x}{3}).$$

```
sage: an = lambda n:4*sin(Pi*n/2)/(Pi *n)
sage: an(1); an(2); an(3)
 1.2732395447351628
 0.000000000000000077963436650387510
 -0.42441318157838759
```

So, $a_1 = \frac{4}{\pi} = 1.273239544735162...$, $a_2 = 0$, $a_3 = -\frac{4}{3\pi}$.

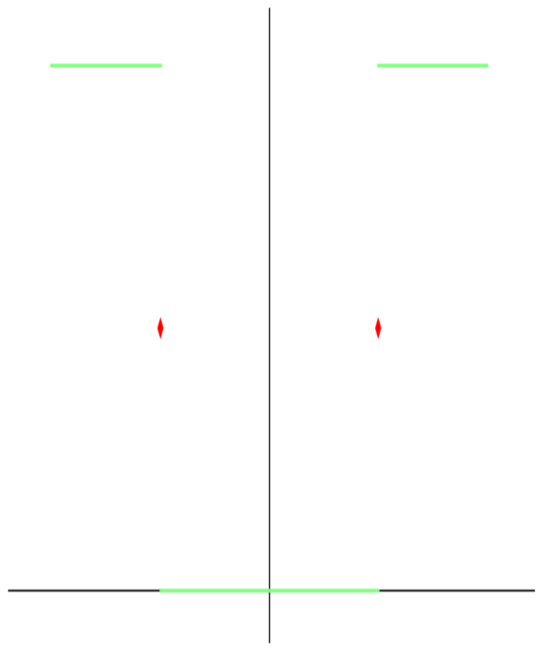


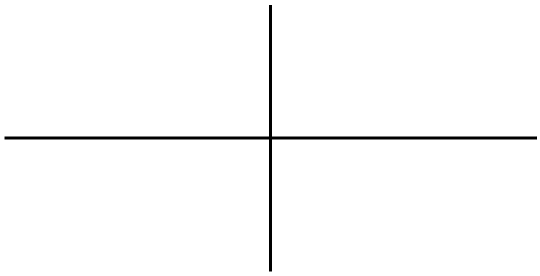
Figure 6: Graph of Cosine series of $f(x)$.

3. Let

$$f(x) = \begin{cases} 2, & 3/2 \le t \le 3, \\ 0, & 0 \le t \le 3/2, \end{cases}$$

- Find the Sine series of $f(x)$ (period 6) and graph the function the FS converges to, $-6 \le x \le 6$.
- Write down the series in summation notation and compute the first 3 non-zero terms.

We leave this one as an exercise for the reader!



2.3 Application to the heat equation

The heat equation with *zero ends* boundary conditions models the temperature of an (insulated) wire of length L:

$$\begin{cases} k\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \\ u(0,t) = u(L,t) = 0. \end{cases}$$

Here $u(x,t)$ denotes the temperature at a point x on the wire at time t. The initial temperature $f(x)$ is specified by the equation

$$u(x,0) = f(x).$$

Method:

- Find the sine series of $f(x)$:

$$f(x) \sim \sum_{n=1}^{\infty} b_n(f) \sin(\frac{n\pi x}{L}),$$

- The solution is

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

Example 14 Let

$$f(x) = \begin{cases} -1, & 0 \le x \le 3/2, \\ 2, & 3/2 < x < 3. \end{cases}$$

Then $L = 3$ and

$$b_n(f) = \frac{2}{3}\int_0^3 f(x)\sin(n\pi x/3)\,dx.$$

Thus

$$f(x) \sim b_1(f)\sin(x\pi/3) + b_2(f)\sin(2x\pi/3) + \ldots = \frac{2}{\pi}\sin(\pi x/3)) - \frac{6}{\pi}\sin(2\pi x/3) + \ldots .$$

The function $f(x)$, and some of the partial sums of its sine series, looks like Figure 7.

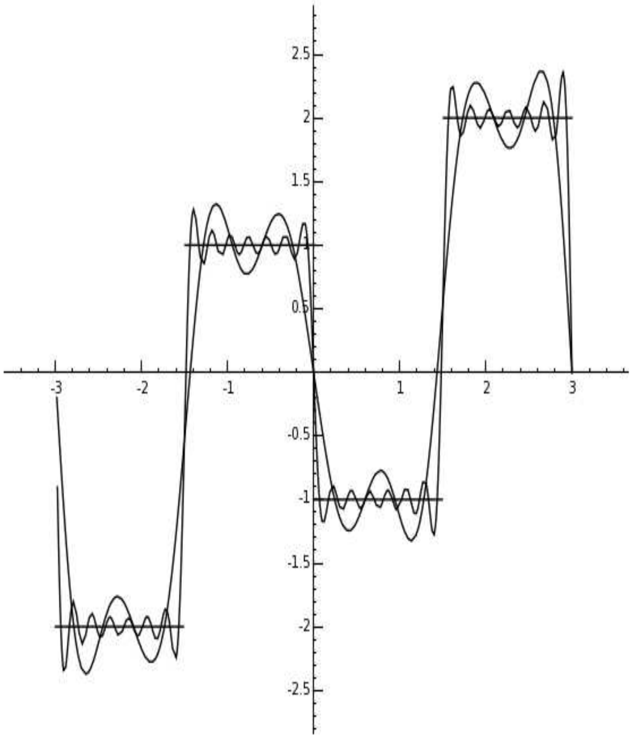


Figure 7: $f(x)$ and two sine series approximations, S_{10}, S_{30}.

As you can see, taking more and more terms gives functions which better and better approximate $f(x)$.

```
sage: R = PolynomialRing(QQ,"x"); x = R.gen()
sage: f1 = -x^0; f2 = 2*x^0
sage: f = Piecewise([[(-3,-3/2),-f2],[(-3/2,0),-f1],[(0,3/2),f1],[(3/2,3),f2]])
sage: fs10 = f.fourier_series_partial_sum(10,3)
sage: FS10 = lambda t:RR(sage_eval(fs10.replace("x",str(t))))
sage: Pfs10 = plot(FS10,-3,3)
sage: Pf = f.plot()
sage: show(Pf+Pfs10)
sage: fs30 = f.fourier_series_partial_sum(30,3)
sage: FS30 = lambda t:RR(sage_eval(fs30.replace("x",str(t))))
sage: Pfs30 = plot(FS30,-3,3)
sage: show(Pf+Pfs10+Pfs30)
```

The solution to the heat equation, therefore, is

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

The heat equation with *insulated ends* boundary conditions models the temperature of an (insulated) wire of length L:

$$\begin{cases} k\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \\ u_x(0,t) = u_x(L,t) = 0. \end{cases}$$

Here $u_x(x,t)$ denotes the partial derivative of the temperature at a point x on the wire at time t. The initial temperature $f(x)$ is specified by the equation $u(x,0) = f(x)$.

Method:

- Find the cosine series of $f(x)$:

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n(f)\cos(\frac{n\pi x}{L}),$$

- The solution is

$$u(x,t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n(f)\cos(\frac{n\pi x}{L}))\exp(-k(\frac{n\pi}{L})^2 t).$$

Example 15 Let

$$f(x) = \begin{cases} -1, & 0 \le x \le 3/2, \\ 2, & \pi/2 < x < 3. \end{cases}$$

Then $L = \pi$ and

$$a_n(f) = \frac{2}{3}\int_0^3 f(x)\cos(n\pi x/3)dx,$$

for $n > 0$ and $a_0 = 1$.

Thus

$$f(x) \sim \frac{a_0}{2} + a_1(f)\cos(x/3) + a_2(f)\cos(2x/3) + ...1/2 - \frac{6}{\pi}\cos(\pi x/3) + \frac{2}{\pi}\cos(3\pi x/3) + ...$$

The piecewise constant function $f(x)$, and some of the partial sums of its cosine series, looks like Figure 8.

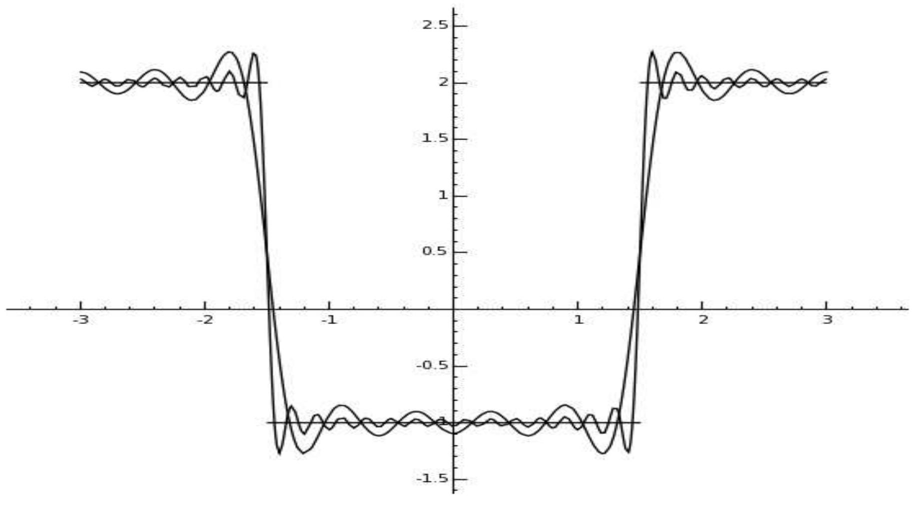


Figure 8: $f(x)$ and two cosine series approximations.

```
sage: R = PolynomialRing(QQ,"x"); x = R.gen()
sage: f1 = -x^0; f2 = 2*x^0
sage: f = Piecewise([[(-3,-3/2),f2],[(-3/2,3/2),f1],[(3/2,3),f2]])
sage: fs10 = f.fourier_series_partial_sum(10,3)
sage: FS10 = lambda t:RR(sage_eval(fs10.replace("x",str(t))))
sage: fs30 = f.fourier_series_partial_sum(30,3)
sage: FS30 = lambda t:RR(sage_eval(fs30.replace("x",str(t))))
sage: Pf = f.plot()
sage: Pfs30 = plot(FS30,-3,3)
sage: Pfs10 = plot(FS10,-3,3)
sage: show(Pf+Pfs10+Pfs30)
```

As you can see, taking more and more terms gives functions which better and better approximate $f(x)$.

The solution to the heat equation, therefore, is

$$u(x,t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n(f)\cos(\frac{n\pi x}{L})\exp(-k(\frac{n\pi}{L})^2 t).$$

Explanation:

Where does this solution come from? It comes from the method of separation of variables and the superposution principle. Here is a short explanation. We shall only discuss the "zero ends" case (the "insulated ends" case is similar).

First, assume the solution to the PDE $k\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t}$ has the "factored" form

$$u(x,t) = X(x)T(t),$$

for some (unknown) functions X, T. If this function solves the PDE then it must satisfy $kX''(x)T(t) = X(x)T'(t)$, or

$$\frac{X''(x)}{X(x)} = \frac{1}{k}\frac{T'(t)}{T(t)}.$$

Since x, t are independent variables, these quotients must be constant. In other words, there must be a constant C such that

$$\frac{T'(t)}{T(t)} = kC, \quad X''(x) - CX(x) = 0.$$

Now we have reduced the problem of solving the one PDE to two ODEs (which is good), but with the price that we have introduced a constant which we don't know, namely C (which maybe isn't so good). The first ODE is easy to solve:

$$T(t) = A_1 e^{kCt},$$

for some constant A_1. To obtain physically meaningful solutions, we do not want the temperature of the wire to become unbounded as time increased (otherwise, the wire would simply melt eventually). Therefore, we may assume here that $C \leq 0$. It is best to analyse two cases now:

Case $C = 0$: This implies $X(x) = A_2 + A_3 x$, for some constants A_2, A_3. Therefore

$$u(x,t) = A_1(A_2 + A_3 x) = \frac{a_0}{2} + b_0 x,$$

where (for reasons explained later) A_1A_2 has been renamed $\frac{a_0}{2}$ and A_1A_3 has been renamed b_0.

Case $C < 0$: Write (for convenience) $C = -r^2$, for some $r > 0$. The ODE for X implies $X(x) = A_2\cos(rx) + A_3\sin(rx)$, for some constants A_2, A_3. Therefore

$$u(x,t) = A_1 e^{-kr^2t}(A_2\cos(rx) + A_3\sin(rx)) = (a\cos(rx) + b\sin(rx))e^{-kr^2t},$$

where A_1A_2 has been renamed a and A_1A_3 has been renamed b.

These are the solutions of the heat equation which can be written in factored form. By superposition, "the general solution" is a sum of these:

$$\begin{aligned} u(x,t) &= \tfrac{a_0}{2} + b_0 x + \textstyle\sum_{n=1}^{\infty}(a_n\cos(r_n x) + b_n\sin(r_n x))e^{-kr_n^2t} \\ &= \tfrac{a_0}{2} + b_0 x + (a_1\cos(r_1 x) + b_1\sin(r_1 x))e^{-kr_1^2t} \\ &\quad +(a_2\cos(r_2 x) + b_2\sin(r_2 x))e^{-kr_2^2t} + ..., \end{aligned} \tag{7}$$

for some a_i, b_i, r_i. We may order the r_i's to be strictly increasing if we like.

We have not yet used the IC $u(x,0) = f(x)$ or the BCs $u(0,t) = u(L,t) = 0$. We do that next.

What do the BCs tell us? Plugging in $x = 0$ into (7) gives

$$0 = u(0,t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n e^{-kr_n^2 t} = \frac{a_0}{2} + a_1 e^{-kr_1^2 t} + a_2 e^{-kr_2^2 t} + \dots .$$

These exponential functions are linearly independent, so $a_0 = 0$, $a_1 = 0$, $a_2 = 0$, ... This implies

$$u(x,t) = b_0 x + \sum_{n=1} b_n \sin(r_n x) e^{-kr_n^2 t} = b_0 x + b_1 \sin(r_1 x) e^{-kr_1^2 t} + b_2 \sin(r_2 x) e^{-kr_2^2 t} + \dots .$$

Plugging in $x = L$ into this gives

$$0 = u(L,t) = b_0 L + \sum_{n=1} b_n \sin(r_n L) e^{-kr_n^2 t}.$$

Again, exponential functions are linearly independent, so $b_0 = 0$, $b_n \sin(r_n L)$ for $n = 1, 2, \dots$. In other to get a non-trivial solution to the PDE, we don't want $b_n = 0$, so $\sin(r_n L) = 0$. This forces $r_n L$ to be a multiple of π, say $r_n = n\pi/L$. This gives

$$u(x,t) = \sum_{n=1}^{\infty} b_n \sin(\frac{n\pi}{L}x) e^{-k(\frac{n\pi}{L})^2 t} = b_1 \sin(\frac{\pi}{L}x)) e^{-k(\frac{\pi}{L})^2 t} + b_2 \sin(\frac{2\pi}{L}x)) e^{-k(\frac{2\pi}{L})^2 t} + \dots, \tag{8}$$

for some b_i's. This was discovered by Fourier.

There is one remaining condition which our solution $u(x,t)$ must satisfy.

What does the IC tell us? Plugging $t = 0$ into (8) gives

$$f(x) = u(x,0) = \sum_{n=1}^{\infty} b_n \sin(\frac{n\pi}{L}x) = b_1 \sin(\frac{\pi}{L}x)) + b_2 \sin(\frac{2\pi}{L}x)) + \dots .$$

In other words, if $f(x)$ is given as a sum of these sine functions, or if we can somehow express $f(x)$ as a sum of sine functions, then we can solve the heat equation. In fact there is a formula for these coefficients b_n (which Fourier did not know at the time):

$$b_n = \frac{2}{L} \int_0^L f(x) \cos(\frac{n\pi}{L}x) dx.$$

It is this formula which is used in the solutions above.

Example 16 Let $k = 1$, let

$$f(x) = \begin{cases} -1, & 0 \le x \le 3/2, \\ 2, & 3/2 < x < 3. \end{cases}$$

and let $g(x) = 0$. Then $L = 3$ and

$$\begin{aligned} b_n(f) &= \tfrac{2}{3} \int_0^3 f(x) \sin(\tfrac{n\pi x}{3}) dx \\ &= -2 \, \tfrac{2 \cos(n\pi) - 3 \cos(1/2 \, n\pi) + 1}{n\pi}. \end{aligned}$$

Thus

$$\begin{aligned} f(x) &\sim b_1(f) \sin(x\pi/3) + b_2(f) \sin(2x\pi/3) + ... \\ &= \tfrac{2}{\pi} \sin(\pi x/3)) - \tfrac{6}{\pi} \sin(2\pi x/3) + ... \, . \end{aligned}$$

The function $f(x)$, and some of the partial sums of its sine series, looks like Figure 7. The solution is

$$\begin{aligned} u(x,t) &= \sum_{n=1}^{\infty} b_n(f) \sin(\tfrac{n\pi x}{3}) \\ &= \tfrac{2}{\pi} \sin(\pi x/3)) e^{-(\pi/3)^2 t} - \tfrac{6}{\pi} \sin(2\pi x/3) e^{-(2\pi/3)^2 t} + ... \, . \end{aligned}$$

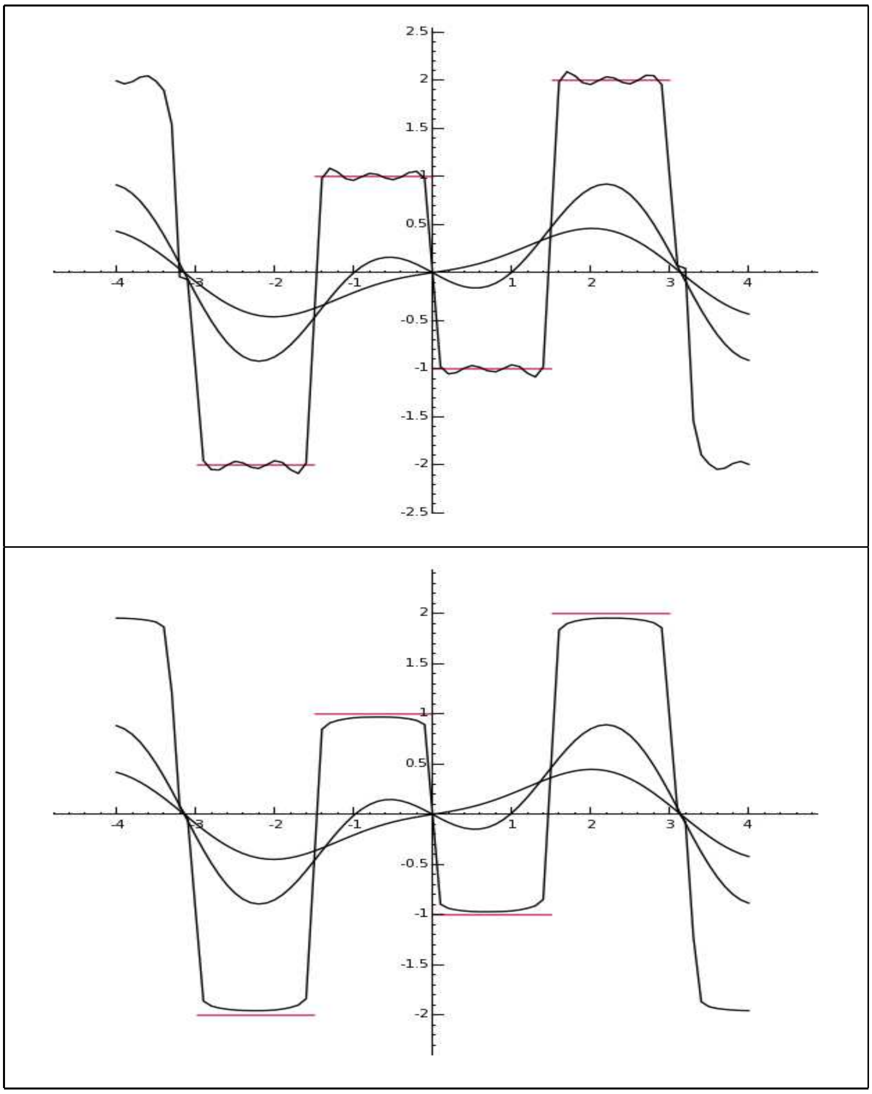


Figure 9: Plot of $f(x)$ and $u(x,t)$, for $t = 0.0, 0.3, 0.6, 0.9$. The first plot uses the partial sums S_{50} of the FS for $f(x)$; the second plot uses the Césaro-filtered partial sums S_{50}^C of the FS for $f(x)$.

The following SAGE code for the plot above is very time-consuming:

```
sage: f1 = lambda x:-2
sage: f2 = lambda x:1
sage: f3 = lambda x:-1
sage: f4 = lambda x:2
```

```
sage: f = Piecewise([[(-3,-3/2),f1],[(-3/2,0),f2],[(0,3/2),f3],[(3/2,3),f4]])
sage: N = 50
sage: t = 0.0; F = [RR(exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: P0 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.3; F = [RR(exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: P1 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.6; F = [RR(exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: P2 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.9; F = [RR(exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: P3 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: P = f.plot(rgbcolor=(0.8,0.1,0.3), plot_points=40)
sage: show(P+P0+P1+P2)

sage: N = 50
sage: t = 0.0; F = [RR((1-j/N)*exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: Pc0 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.3; F = [RR((1-j/N)*exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: Pc1 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.6; F = [RR((1-j/N)*exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: Pc2 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.9; F = [RR((1-j/N)*exp(-(j*pi/3)^2*t)) for j in range(N)]
sage: Pc3 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: show(P+Pc0+Pc1+Pc2)
```

2.4 Application to Schödinger's equation

The one-dimensional Schrödinger equation for a free particle is

$$ik\frac{\partial^2\psi(x,t)}{\partial x^2} = \frac{\partial\psi(x,t)}{\partial t},$$

where $k > 0$ is a constant (involving Planck's constant and the mass of the particle) and $i = \sqrt{-1}$ as usual. The solution ψ is called the *wave function* describing instantaneous "state" of the particle. For the analog in 3 dimensions (which is the one actually used by physicists - the one-dimensional version we are dealing with is a simplified mathematical model), one can interpret the square of the absolute value of the wave function as the probability density function for the particle to be found at a point in space. In other words, $|\psi(x,t)|^2\, dx$ is the probability of finding the particle in the "volume dx" surrounding the position x, at time t.

If we restrict the particle to a "box" then (for our simplied one-dimensional quantum-mechanical model) we can impose a boundary condition of the form

$$\psi(0,t) = \psi(L,t) = 0,$$

and an initial condition of the form

$$\psi(x,0) = f(x), \qquad 0 < x < L.$$

Here f is a function (sometimes simply denoted $\psi(x)$) which is normalized so that

$$\int_0^L |f(x)|^2\, dx = 1.$$

If $|\psi\,(x,t)|^2$ represents a pdf of finding a particle "at x" at time t then $\int_0^L |f(x)|^2\, dx$ represents the probability of finding the particle somewhere in the "box" initially, which is of course 1.

Method:

- Find the sine series of $f(x)$:

$$f(x) \sim \sum_{n=1}^{\infty} b_n(f)\sin(\frac{n\pi x}{L}),$$

- The solution is

$$\psi(x,t) = \sum_{n=1}^{\infty} b_n(f)\sin(\frac{n\pi x}{L})\exp(-ik(\frac{n\pi}{L})^2 t).$$

Each of the terms

$$\psi_n(x,t) = b_n \sin(\frac{n\pi x}{L})\exp(-ik(\frac{n\pi}{L})^2 t).$$

is called a *standing wave* (though in this case sometimes b_n is chosen so that $\int_0^L |\psi_n(x,t)|^2\, dx = 1$).

Example 17 Let

$$f(x) = \begin{cases} -1, & 0 \le x \le 1/2, \\ 1, & 1/2 < x < 1. \end{cases}$$

Then $L = 1$ and

$$b_n(f) = \frac{2}{1}\int_0^1 f(x)\sin(\frac{n\pi x}{1})dx = \frac{1}{n\pi}(-1 + 2\cos(\frac{n\pi}{2}) - \cos(n\pi)).$$

Thus

$$\begin{array}{ll} f(x) & \sim b_1(f)\sin(\pi x) + b_2(f)\sin(2\pi x) + ... \\ & = \sum_n \frac{1}{n\pi}(-1 + 2\cos(\frac{n\pi}{2}) - \cos(n\pi)) \cdot \sin(n\pi x) \ . \end{array}$$

Taking more and more terms gives functions which better and better approximate $f(x)$. The solution to Schrödinger's equation, therefore, is

$$\psi(x,t) = \sum_{n=1}^{\infty} \frac{1}{n\pi}(-1 + 2\cos(\frac{n\pi}{2}) - \cos(n\pi)) \cdot \sin(n\pi x) \cdot \exp(-ik(n\pi)^2 t).$$

Explanation:

Where does this solution come from? It comes from the method of separation of variables and the superposution principle. Here is a short explanation.

First, assume the solution to the PDE $ik\frac{\partial^2\psi(x,t)}{\partial x^2} = \frac{\partial\psi(x,t)}{\partial t}$ has the "factored" form

$$\psi(x,t) = X(x)T(t),$$

for some (unknown) functions X, T. If this function solves the PDE then it must satisfy $kX''(x)T(t) = X(x)T'(t)$, or

$$\frac{X''(x)}{X(x)} = \frac{1}{ik}\frac{T'(t)}{T(t)}.$$

Since x, t are independent variables, these quotients must be constant. In other words, there must be a constant C such that

$$\frac{T'(t)}{T(t)} = ikC, \quad X''(x) - CX(x) = 0.$$

Now we have reduced the problem of solving the one PDE to two ODEs (which is good), but with the price that we have introduced a constant which we don't know, namely C (which maybe isn't so good). The first ODE is easy to solve:

$$T(t) = A_1 e^{ikCt},$$

for some constant A_1. It remains to "determine" C.

Case $C > 0$: Write (for convenience) $C = r^2$, for some $r > 0$. The ODE for X implies $X(x) = A_2 \exp(rx) + A_3 \exp(-rx)$, for some constants A_2, A_3. Therefore

$$\psi(x,t) = A_1 e^{-ikr^2 t}(A_2 \exp(rx) + A_3 \exp(-rx)) = (a \exp(rx) + b \exp(-rx)) e^{-ikr^2 t},$$

where $A_1 A_2$ has been renamed a and $A_1 A_3$ has been renamed b. This will not match the boundary conditions unless a and b are both 0.

Case $C = 0$: This implies $X(x) = A_2 + A_3 x$, for some constants A_2, A_3. Therefore

$$\psi(x,t) = A_1(A_2 + A_3 x) = a + bx,$$

where $A_1 A_2$ has been renamed a and $A_1 A_3$ has been renamed b. This will not match the boundary conditions unless a and b are both 0.

Case $C < 0$: Write (for convenience) $C = -r^2$, for some $r > 0$. The ODE for X implies $X(x) = A_2 \cos(rx) + A_3 \sin(rx)$, for some constants A_2, A_3. Therefore

$$\psi(x,t) = A_1 e^{-ikr^2 t}(A_2 \cos(rx) + A_3 \sin(rx)) = (a \cos(rx) + b \sin(rx)) e^{-ikr^2 t},$$

where $A_1 A_2$ has been renamed a and $A_1 A_3$ has been renamed b. This will not match the boundary conditions unless $a = 0$ and $r = \frac{n\pi}{L}$

These are the solutions of the heat equation which can be written in factored form. By superposition, "the general solution" is a sum of these:

$$\begin{aligned} \psi(x,t) &= \sum_{n=1}^{\infty} (a_n \cos(r_n x) + b_n \sin(r_n x)) e^{-ikr_n^2 t} \\ &= b_1 \sin(r_1 x) e^{-ikr_1^2 t} + b_2 \sin(r_2 x) e^{-ikr_2^2 t} + ..., \end{aligned} \tag{9}$$

for some b_n, where $r_n = \frac{n\pi}{L}$. Note the similarity with Fourier's solution to the heat equation.

There is one remaining condition which our solution $\psi(x,t)$ must satisfy. We have not yet used the IC $\psi(x,0) = f(x)$. We do that next.

Plugging $t = 0$ into (9) gives

$$f(x) = \psi(x,0) = \sum_{n=1}^{\infty} b_n \sin(\frac{n\pi}{L}x) = b_1 \sin(\frac{\pi}{L}x)) + b_2 \sin(\frac{2\pi}{L}x)) + ... \ .$$

In other words, if $f(x)$ is given as a sum of these sine functions, or if we can somehow express $f(x)$ as a sum of sine functions, then we can solve Schrödinger's equation. In fact there is a formula for these coefficients b_n:

$$b_n = \frac{2}{L}\int_0^L f(x)\cos(\frac{n\pi}{L}x)dx.$$

It is this formula which is used in the solutions above.

2.5 Application to the wave equation

The wave equation with zero ends boundary conditions models the motion of a (perfectly elastic) guitar string of length L:

$$\begin{cases} \alpha^2 \frac{\partial^2 w(x,t)}{\partial x^2} = \frac{\partial^2 w(x,t)}{\partial t^2} \\ w(0,t) = w(L,t) = 0. \end{cases}$$

Here $w(x,t)$ denotes the displacement from rest of a point x on the string at time t. The initial displacement $f(x)$ and initial velocity $g(x)$ are specified by the equations

$$w(x,0) = f(x), \qquad w_t(x,0) = g(x).$$

Method:

- Find the sine series of $f(x)$ and $g(x)$:

$$f(x) \sim \sum_{n=1}^{\infty} b_n(f)\sin(\frac{n\pi x}{L}), \qquad g(x) \sim \sum_{n=1}^{\infty} b_n(g)\sin(\frac{n\pi x}{L}).$$

- The solution is

$$w(x,t) = \sum_{n=1}^{\infty}(b_n(f)\cos(\frac{\alpha n\pi t}{L}) + \frac{Lb_n(g)}{n\pi\alpha}\sin(\frac{\alpha n\pi t}{L}))\sin(\frac{n\pi x}{L}).$$

A special case: When there is no initial velocity then $g = 0$ and the solution to the wave equation, therefore, is

$$w(x,t) = \sum_{n=1}^{\infty} b_n(f)\cos(\frac{\alpha n\pi t}{L})\sin(\frac{n\pi x}{L}).$$

Example 18 Let $\alpha = 1$, let

$$f(x) = \begin{cases} -1, & 0 \leq x \leq \pi/2, \\ 2, & \pi/2 < x < \pi. \end{cases}$$

and let $g(x) = 0$. Then $L = \pi$, $b_n(g) = 0$, and

$$\begin{aligned} b_n(f) &= \tfrac{2}{\pi}\int_0^{\pi} f(x)\sin(nx)dx \\ &= -2\,\tfrac{2\,\cos(n\pi)-3\,\cos(1/2\,n\pi)+1}{n\pi}. \end{aligned}$$

Thus

$$\begin{aligned} f(x) &\sim b_1(f)\sin(x) + b_2(f)\sin(2x) + ... \\ &= \tfrac{2}{\pi}\sin(x) - \tfrac{6}{\pi}\sin(2x) + \tfrac{2}{3\pi}\sin(3x) + ... \end{aligned}$$

The function $f(x)$, and some of the partial sums of its sine series, looks like Figure 7. The solution is

$$\begin{aligned} w(x,t) &= \textstyle\sum_{n=1}^{\infty} b_n(f)\cos(nt)\sin(nx) \\ &= \tfrac{2}{\pi}\cos(t)\sin(x) - \tfrac{6}{\pi}\cos(2t)\sin(2x) + \tfrac{2}{3\pi}\cos(3t)\sin(3x) + ...\ . \end{aligned}$$

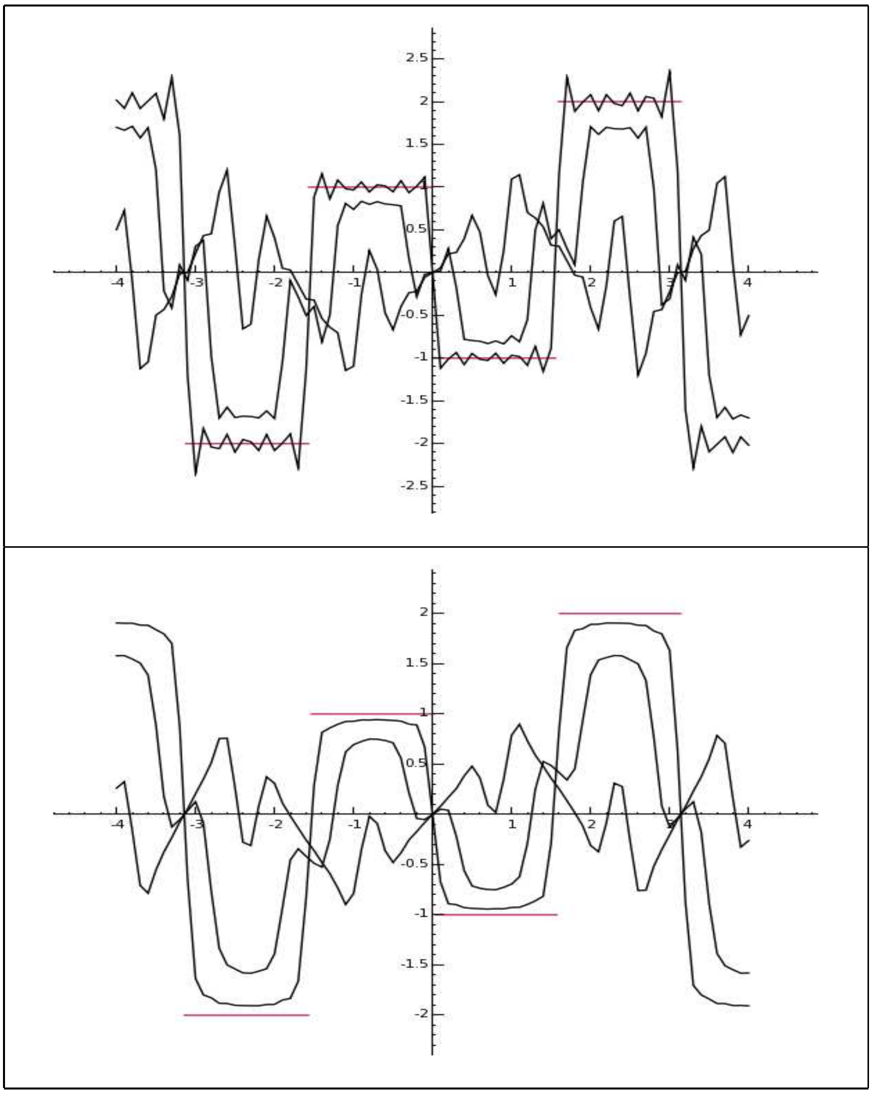


Figure 10: Plot of $f(x)$ and $w(x,t)$, for $t = 0.0, 0.3, 0.6, 0.9$. The first plot uses the partial sums S_{25} of the FS for $f(x)$; the second plot uses the Césaro-filtered partial sums S_{25}^C of the FS for $f(x)$.

The following SAGE code for the plot above is very time-consuming:

```
sage: f1 = lambda x:-2
sage: f2 = lambda x:1
sage: f3 = lambda x:-1
sage: f4 = lambda x:2
```

```
sage: f = Piecewise([[(-pi,-pi/2),f1],[(-pi/2,0),f2],[(0,pi/2),f3],[(pi/2,pi),f4]])
sage: N = 25
sage: t = 0.0; F = [RR(cos((j+1)*t)) for j in range(N)]
sage: P0 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.3; F = [RR(cos((j+1)*t)) for j in range(N)]
sage: P1 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.6; F = [RR(cos((j+1)*t)) for j in range(N)]
sage: P2 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.9; F = [RR(cos((j+1)*t)) for j in range(N)]
sage: P3 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: P = f.plot(rgbcolor=(0.8,0.1,0.3), plot_points=40)
sage: show(P+P0+P1+P2)

sage: f1 = lambda x:-2
sage: f2 = lambda x:1
sage: f3 = lambda x:-1
sage: f4 = lambda x:2
sage: f = Piecewise([[(-pi,-pi/2),f1],[(-pi/2,0),f2],[(0,pi/2),f3],[(pi/2,pi),f4]])
sage: N = 25
sage: t = 0.0; F = [RR((1-j/N)*cos((j+1)*t)) for j in range(N)]
sage: P0 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.3; F = [RR((1-j/N)*cos((j+1)*t)) for j in range(N)]
sage: P1 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.6; F = [RR((1-j/N)*cos((j+1)*t)) for j in range(N)]
sage: P2 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: t = 0.9; F = [RR((1-j/N)*cos((j+1)*t)) for j in range(N)]
sage: P3 = f.plot_fourier_series_partial_sum_filtered(N,pi,F,-4,4)
sage: P = f.plot(rgbcolor=(0.8,0.1,0.3), plot_points=40)
sage: show(P+P0+P1+P2)
```

3 The Discrete Fourier transform

Let us first "discretize" the integral for the k-th coefficient of the Fourier series and use that as a basis for defining the DFT. Using the "left-hand Riemann sum" approximation for the integral using N subdivisions, we have

$$\begin{aligned} c_k &= \tfrac{1}{P}\int_0^P f(x)e^{-2\pi ikx/P}\,dx \\ &\approx \tfrac{1}{P}\sum_{j=0}^{N-1} f(Pj/N)e^{-2\pi ik\frac{Pj}{N}/P}\left(\tfrac{P}{N}\right) \\ &= \tfrac{1}{N}\sum_{j=0}^{N-1} f(Pj/N)e^{-2\pi ikj/N}. \end{aligned} \tag{10}$$

This motivates the following definition.

Definition 19 The *N-point discrete Fourier transform* (or *DFT*) of the vector $\vec{f} = (f_0, ..., f_{N-1}) \in \mathbb{C}^N$ is

$$DFT(\vec{f})_k = \hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-2\pi ikj/N} = \sum_{j=0}^{N-1} f_j \overline{W}^{kj},$$

where $W = e^{2\pi i/N}$.

The *normalized N-point discrete Fourier transform* (or *NDFT*) of the vector $\vec{f} = (f_0, ..., f_{N-1}) \in \mathbb{C}^N$ is

$$NDFT(\vec{f})_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_j e^{-2\pi ikj/N} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_j \overline{W}^{kj}.$$

Note that the powers of W are N equally distributed points on the unit circle.

Both the DFT and NDFT define linear transformations $\mathbb{C}^N \to \mathbb{C}^N$ and therefore can be described by matrices. If we regard the vector $\vec{f}$ as a column vector then the matrix for the DFT is:

$$F_N = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \overline{W} & \dots & \overline{W}^{N-1} \\ 1 & \overline{W}^2 & \dots & \overline{W}^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \overline{W}^{N-1} & \dots & \overline{W}^{(N-1)(N-1)} \end{pmatrix}.$$

Note that this is a symmetric matrix. Similarly, the matrix for the NDFT is:

$$G_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \overline{W} & \dots & \overline{W}^{N-1} \\ 1 & \overline{W}^2 & \dots & \overline{W}^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \overline{W}^{N-1} & \dots & \overline{W}^{(N-1)(N-1)} \end{pmatrix}.$$

Example 20 *Let $N = 10$. The DFT of*

$$\vec{f} = (1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10) \in \mathbb{C}^{10}$$

is

$F_{10}\vec{f} = (1,0,0,0,0,0,0,0,0,0)$. *The DFT of*

$$\vec{f} = (1/10,0,0,0,0,0,0,0,0,0) \in \mathbb{C}^{10}$$

is $F_{10}\vec{f} = (1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10)$.

This was computed using the SAGE commands

```
sage: J = range(10)
sage: A = [1/10 for j in J]
sage: s = IndexedSequence(A,J)
sage: s.dft()
Indexed sequence: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    indexed by [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

There is an analog of the orthogonality used in Theorem 4 above.

Lemma 21 *We have*

$$\sum_{j=0}^{N-1} \overline{W}^{kj} = \begin{cases} N, & N|k, \\ 0, & \text{otherwise.} \end{cases}$$

Before proving this algebraically, I claim that this is "geometrically obvious." To see this, recall that the average of any N points in the plane - whether written as vectors or as complex numbers - is simply the "center of gravity" of points, regarded as equally weighted point masses. The sum above is (if N does not divide k) the "center of gravity" of a collection of point masses which are equi-distributed about the unit circle. This center of gravity must be the origin. On the other hand, if $N|k$ then all the points are concentrated at 1, so the total mass is N in that case.

proof: If $\overline{W}^k \neq 1$ then we have

$$\sum_{j=0}^{N-1} \overline{W}^{kj} = \frac{\overline{W}^{Nk} - 1}{\overline{W}^k - 1} = 0.$$

If $\overline{W}^k = 1$ then we $\sum_{j=0}^{N-1} \overline{W}^{kj} = N$. Note $\overline{W}^k = 1$ if and only if $N|k$. $\Box$

As a corollary of this lemma, we see that the complex matrix F_N is "orthogonal" (this is a technical term we need to define). A real square matrix is called *orthogonal* if row i is orthogonal to column j for all $i \neq j$. Here when we say two real vectors are orthogonal of course we mean that they are non-zero vectors and that their dot product is 0. The definition for complex matrices is a bit different. We first define on $\mathbb{C}^N$ the *Hermitian inner product* (or just *inner product*, for short):

$$\langle \vec{x}, \vec{y} \rangle = \sum_{j=0}^{N-1} x_j \overline{y_j}.$$

We say two complex vectors are *orthogonal* if they are non-zero and their inner product is zero. A complex square matrix is called *orthogonal* if row i is orthogonal to column j for all $i \neq j$.

Lemma 22 *F_N is orthogonal.*

proof: The k-th row of F_N is the vector $(\overline{W}^{(k-1)j})_{j=0,\ldots,N-1}$, and the complex conjugate of this vector is the vector $(W^{(k-1)j})_{j=0,\ldots,N-1} = (\overline{W}^{-(k-1)j})_{j=0,\ldots,N-1}$, so

$$\langle (\mathrm{row\ k\ of\ F_N}), (\mathrm{row\ } \ell \mathrm{\ of\ F_N}) \rangle = \sum_{j=0}^{N-1} \overline{W}^{((k-1)-(\ell-1))j} = 0,$$

provided $\overline{W}^{k-\ell} \neq 1$, which is true if and only if N does not divide $k - \ell$. $\Box$

Note that this matrix F_N is not "real orthogonal":

$$(\mathrm{row\ k\ of\ F_N}) \cdot (\mathrm{row\ } \ell \mathrm{\ of\ F_N}) = \sum_{j=0}^{N-1} \overline{W}^{(k-1)+(\ell-1)j} = 0,$$

if and only if N does not divide $k + \ell - 2$, by Lemma 21.

Here's another matrix calculation based on Lemma 21: If $\vec{e}_k$ denotes the standard basis vector of $\mathbb{C}^N$ whose k-th coordinate is 1 and all other coordinates are 0, then

$$DFT(\vec{e}_k) = (F_N)_k = k^{th} \text{ column of } F_N,$$

so

$$
\begin{aligned}
DFT(DFT(\vec{e}_k)) &= DFT(k^{th} \text{ column of } F_N), \\
&= \begin{pmatrix} (1^{st} \text{ row of } F_N) \cdot (k^{th} \text{ column of } F_N) \\ (2^{nd} \text{ row of } F_N) \cdot (k^{th} \text{ column of } F_N) \\ \vdots \\ (N^{th} \text{ row of } F_N) \cdot (k^{th} \text{ column of } F_N) \end{pmatrix} \\
&= \begin{pmatrix} (1^{st} \text{ row of } F_N) \cdot (k^{th} \text{ row of } F_N) \\ (2^{nd} \text{ row of } F_N) \cdot (k^{th} \text{ row of } F_N) \\ \vdots \\ (N^{th} \text{ row of } F_N) \cdot (k^{th} \text{ row of } F_N) \end{pmatrix},
\end{aligned}
$$

since the matrix F_N is symmetric. The "almost orthogonality of the rows of F_N" discussed above implies that this last vector of dot products is $= N\vec{e}_{-k}$, where by $-k$ in the subscript of we mean the representative of the residue class of $-k \pmod N$ in the interval $0 \leq -k \leq N-1$.

The motivation behind the definition of the normalized DFT is the following computation:

$$
\begin{aligned}
NDFT(NDFT(\vec{f}))_k &= \tfrac{1}{\sqrt{N}} \textstyle\sum_{j=0}^{N-1} NDFT(f)_j \overline{W}^{kj} \\
&= \tfrac{1}{N} \textstyle\sum_{\ell=0}^{N-1} f_\ell \sum_{j=0}^{N-1} \overline{W}^{(k+\ell)j} \qquad (11) \\
&= f_{-k},
\end{aligned}
$$

where of course by $-k$ in the subscript of f_{-k} we mean the representative of the residue class of $-k \pmod N$ in the interval $0 \leq -k \leq N-1$. To be precise, if $\mathrm{neg} : \mathbb{C}^N \to \mathbb{C}^N$ denotes the negation operator, sending $(f_0, f_1, ..., f_{N-1})$ to $(f_{-0}, f_{-1}, ..., f_{1-N})$ then $NDFT^2 = \mathrm{neg}$. Note that neg flips the last $N-1$ coordinates of $\vec{f}$ about their midpoint. For example, if $N = 5$ then $\mathrm{neg}(1, 2, 3, 4, 5) = (1, 5, 4, 3, 2)$ and if $N = 6$ then $\mathrm{neg}(2, 1, 3, -1, 4, 7) = (2, 7, 4, -1, 3, 1)$.

Because of the computation (11), it follows that

$$
NDFT^{-1}(\vec{f})_k = NDFT(\vec{f})_{-k} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_j \overline{W}^{-kj} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_j W^{kj}, \quad (12)
$$

or $NDFT^{-1} = \text{neg} \circ NDFT = NDFT \circ \text{neg}$. Likewise,

$$DFT^{-1}(\vec{f})_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j \overline{W}^{-kj} = \frac{1}{N}\sum_{j=0}^{N-1} f_j W^{kj}, \tag{13}$$

or $DFT^{-1} = N^{-1} \cdot \text{neg} \circ DFT = N^{-1} \cdot DFT \circ \text{neg}$.

Example 23 *Let $N = 4$, so*

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix},$$

and let $\vec{f} = (1, 0, 0, 1)$. We compute

$$NDFT(\vec{f}) = \frac{1}{\sqrt{4}} F_4 \vec{f} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ (1+i)/2 \\ 0 \\ (1-i)/2 \end{pmatrix}.$$

Call this latter vector $\vec{g}$. We compute

$$NDFT \circ \text{neg}(\vec{g}) = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}\begin{pmatrix} 1 \\ (1+i)/2 \\ 0 \\ (1-i)/2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix} = \vec{f},$$

as desired.

3.1 Eigenvalues and eigenvectors of the DFT

We shall take an aside to try to address the problem of computing eigenvalues and eigenvectors of $DFT = F_N$, at least in a special case. This shall not be used in other parts of the course - think of this as "for your cultural benefit".

It also follows from this computation (11) that $NDFT^4$ acts as the identity. If A is any square matrix satisfying $A^m = I$ then any eigenvalue λ of A must be an m-th root of unity (indeed, $A\vec{v} = \lambda\vec{v}$, for some non-zero eigenvector $\vec{v}$, so $\vec{v} = A^m\vec{v} = A^{m-1}A\vec{v} = A^{m-1}\lambda\vec{v} = \lambda A^{m-1}\vec{v} = ... = \lambda^m\vec{v}$, so $\lambda^m = 1$). Due to this fact, it follows that the only possible eigenvalues of $NDFT$ are $1, -1, i, -i$.

It is not hard to describe the eigenspaces intrinsically. Let

$$V = \{f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{C}\},$$

so $V \cong \mathbb{C}^N$ via $f \longmapsto (f(0), f(1), ..., f(N-1))$. Let

$$V_{even} = \{f \in V \mid f(-k) = f(k),\ \forall k \in \mathbb{Z}/N\mathbb{Z}\}$$

denote the subspace of even functions and

$$V_{odd} = \{f \in V \mid f(-k) = -f(k),\ \forall k \in \mathbb{Z}/N\mathbb{Z}\}$$

the subspace of odd functions. Note that the restriction of $NDFT$ to V_{even} is order 2: for all $f \in V_{even}$, we have $NDFT^2(f) = f$. Likewise, for all $f \in V_{odd}$, we have $NDFT^2(f) = -f$. Let

$$\begin{aligned}
E_1 &= \{NDFT(f) + f \mid f \in V_{even}\},\\
E_{-1} &= \{NDFT(f) - f \mid f \in V_{even}\},\\
E_{-i} &= \{iNDFT(f) + f \mid f \in V_{odd}\},\\
E_i &= \{iNDFT(f) - f \mid f \in V_{odd}\}.
\end{aligned}$$

In this notation, for each $\lambda \in \{\pm1, \pm i\}$, E_λ is the eigenspace of G_N having eigenvalue λ. Moreover, according to Good[5] [G], the columns of the matrix

$$M_\lambda = I + \lambda G_N + \lambda^2 G_N^2 + \lambda^3 G_N^3 \tag{14}$$

form a basis for E_λ.

Note that the eigenvalues of the $DFT = F_N$ are not the same as those of $NDFT = G_N$ since $G_N = \frac{1}{\sqrt{N}}F_N$. The eigenvalues of the DFT must belong to $\sqrt{N}, -\sqrt{N}, i\sqrt{N}, -i\sqrt{N}$.

Recall that for the Fourier transform on $\mathbb{R}$ the number $\lambda = 1$ was an eigenvalue. The calculation there cannot be "discretized" since we had to

[5]Good's definition of the NDFT is slightly different than ours. Essentially, where we have a weighted sum over powers of $\overline{W}$, he has a weighted sum over powers of W. That changes the matrix M_λ slightly, so the one above is correct for us I think.

use the Residue Theorem from complex analysis. We have to take another approach.

Let $\mathbb{Z}/N\mathbb{Z} = \{0, 1, 2, ..., N-1\}$. This is a set but if we perform arithmetic (such as addition and multiplication) mod N, then this can regarded as an abelian group.

First, assume that there is a function

$$\ell : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z} \tag{15}$$

with the property that $\ell(jk) = \ell(j)\ell(k)$, for all $j \neq 0$ and $k \neq 0$, and $\ell(0) = 0$. Let us also assume that the set

$$k\mathbb{Z}/N\mathbb{Z} = \{jk \pmod N \mid j = 0, 1, ..., N-1\}$$

which is a subset of $\mathbb{Z}/N\mathbb{Z}$, is the same as the set $\mathbb{Z}/N\mathbb{Z}$:

$$k\mathbb{Z}/N\mathbb{Z} = \mathbb{Z}/N\mathbb{Z}, \quad 0 < k < N. \tag{16}$$

We will worry about whether this function exists or not and whether this set-theoretic property of true or not later. For now, let's compute the first component of it's DFT:

$$FDT(\vec{\ell})_1 = \sum_{j=0}^{N-1} g(j) e^{-2\pi i j/N},$$

where $\vec{\ell} = (\ell(0), \ell(1), ..., \ell(N-1)) = (0, \ell(1), ..., \ell(N-1))$. Now make the substitution $j = j'k$, for some $k \neq 0$.

We have

$$\begin{array}{ll} FDT(\vec{\ell})_1 & = \sum_{j=0}^{N-1} \ell(j) e^{-2\pi i j/N} \\ & = \sum_{j'=0}^{N-1} \ell(j'k) e^{-2\pi i j'k/N} \\ & = \sum_{j'=0}^{N-1} \ell(j')\ell(k) e^{-2\pi i j'k/N} \\ & = \ell(k) \sum_{j=0}^{N-1} \ell(j) e^{-2\pi i jk/N} \\ & = \ell(k) DFT(\vec{\ell})_k. \end{array}$$

Putting these together, we get

$$DFT(\vec{\ell}) = DFT(\vec{\ell})_1 \cdot \vec{\ell}.$$

In other words, $\vec{\ell}$ is an eigenvector with eigenvalue $DFT(\vec{\ell})_1$.

Example 24 *In general. let ζ_N denote a primitive N-th root of unity, for example $\zeta_N = e^{-2\pi i/N} = \overline{W}$. The smallest field containing the rationals, $\mathbb{Q}$, and this N-th root of unity is called the cyclotomic field, and is commonly denoted by $\mathbb{Q}(\zeta_N)$. As a set, $\mathbb{Q}(\zeta_N)$ is simply the set of "polynomials in ζ_N with coefficients in $\mathbb{Q}$ of degree $\leq N-1$".*

Let $N = 5$ and $\vec{\ell} = (0, 1, -1, -1, 1)$. It can be checked that this defines a function as in (15).

We first check explicitly that assumption (16) is true:

$$\begin{array}{c} \mathbb{Z}/5\mathbb{Z} = [0,1,2,3,4] \\ 2\mathbb{Z}/5\mathbb{Z} = [0,2,4,1,3] \\ 3\mathbb{Z}/5\mathbb{Z} = [0,3,1,4,2] \\ 4\mathbb{Z}/5\mathbb{Z} = [0,4,3,2,1] \end{array}$$

Let

$$F_5 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta_5 & \zeta_5^2 & \zeta_5^3 & \zeta_5^4 \\ 1 & \zeta_5^2 & \zeta_5^4 & \zeta_5 & \zeta_5^3 \\ 1 & \zeta_5^3 & \zeta_5 & \zeta_5^4 & \zeta_5^2 \\ 1 & \zeta_5^4 & \zeta_5^3 & \zeta_5^2 & \zeta_5 \end{pmatrix}$$

where $\zeta_5^4 = -\zeta_5^3 - \zeta_5^2 - \zeta_5 - 1$. The characteristic polynomial of this matrix is

$$p_5(x) = (x - \sqrt{5})(x + \sqrt{5})^2(x^2 + 5).$$

The roots of this polynomial are of course the eigenvalues: $\pm\sqrt{5}, \pm i\sqrt{5}$. Of course, the matrix F_5 has eigenvectors with vector components in $\mathbb{C}$. We shall try to express their components, if we can, algebraically in terms of the powers of the ζ_5. This is not a matter of necessity for us, but it can be convenient for doing certain calculations. For example, you don't have to worry about round-off errors messing up a calculation if you have an algebraic expression.

It turns out that $\sqrt{5} = -2\zeta_5^3 - 2\zeta_5^2 - 1$, and $-\sqrt{5} = 2\zeta_5^3 + 2\zeta_5^2 + 1$, can both be written as a linear combination of powers of ζ_5, so it may come as no surprise that the components of their eigenvectors also can be written as a linear combination of powers of ζ_5. In other words, if the eigenvalue is in $\mathbb{Q}(\zeta_5)$ then one might expect to find eigenvector with components in $\mathbb{Q}(\zeta_5)$.

On the other hand, $\pm i\sqrt{5}$ do not belong to $\mathbb{Q}(\zeta_5)$. So, we should expect that the eigenvectors in this case have components lying in some extension of *in $\mathbb{Q}(\zeta_5)$. It turns out, all the eigenvectors have components in $\mathbb{Q}(\zeta_{20})$, the field extension of $\mathbb{Q}$ generated by the 20-th roots of unity.*

The eigenspace of $\lambda_1 = \sqrt{5}$ is 2-dimensional, with basis

$$(1, 0, -\zeta_5^3 - \zeta_5^2 - 1, -\zeta_5^3 - \zeta_5^2 - 1, 0), \quad (0, 1, -1, -1, 1),$$

where $-\zeta_5^3 - \zeta_5^2 - 1 = 0.6180...$. The eigenspace of $\lambda_2 = -\sqrt{5}$ is 1-dimensional, with basis

$$(1, \frac{1}{2}\zeta_5^3 + \frac{1}{2}\zeta_5^2, \frac{1}{2}\zeta_5^3 + \frac{1}{2}\zeta_5^2, \frac{1}{2}\zeta_5^3 + \frac{1}{2}\zeta_5^2, \frac{1}{2}\zeta_5^3 + \frac{1}{2}\zeta_5^2),$$

where $\frac{1}{2}\zeta_5^3 + \frac{1}{2}\zeta_5^2 = -0.8090...$. The eigenspace of $\lambda_3 = i\sqrt{5}$ is 1-dimensional, with basis

$$(0, 1, \zeta_{20}^7 + \zeta_{20}^6 - \zeta_{20}^5 - \zeta_{20}^4 + \zeta_{20}^3 - 2\zeta_{20} - 1, -\zeta_{20}^7 - \zeta_{20}^6 + \zeta_{20}^5 + \zeta_{20}^4 - \zeta_{20}^3 + 2\zeta_{20} + 1, -1),$$

where $\zeta_{20}^7 + \zeta_{20}^6 - \zeta_{20}^5 - \zeta_{20}^4 + \zeta_{20}^3 - 2\zeta_{20} - 1 = -3.520...$. The eigenspace of $\lambda_4 = i\sqrt{5}$ is 1-dimensional, with basis

$$(0, 1, -\zeta_{20}^7 + \zeta_{20}^6 + \zeta_{20}^5 - \zeta_{20}^4 - \zeta_{20}^3 + 2\zeta_{20} - 1, \zeta_{20}^7 - \zeta_{20}^6 - \zeta_{20}^5 + \zeta_{20}^4 + \zeta_{20}^3 - 2\zeta_{20} + 1, -1),$$

where $-\zeta_{20}^7 + \zeta_{20}^6 + \zeta_{20}^5 - \zeta_{20}^4 - \zeta_{20}^3 + 2\zeta_{20} - 1 = 0.2840...$.

For example,

$$F_5 \vec{\ell} = F_5 \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2\zeta_5^3 - 2\zeta_5^2 - 1 \\ 2\zeta_5^3 + 2\zeta_5^2 + 1 \\ 2\zeta_5^3 + 2\zeta_5^2 + 1 \\ -2\zeta_5^3 - 2\zeta_5^2 - 1 \end{pmatrix} = (2\zeta_5^3 + 2\zeta_5^2 + 1)\vec{\ell}.$$

In fact, $2\zeta_5^3 + 2\zeta_5^2 + 1 = -\sqrt{5} \approx -2.236...$. In other words, $\vec{\ell} = (0, 1, -1, -1, 1)$ is an eigenvector of F_5 with eigenvalue $\lambda = -\sqrt{5}$.

The `SAGE` *[S] code used to help with this calculation:*

```
----------------------------------------------------------------------
| SAGE Version 1.7.1, Release Date: 2007-01-18                       |
| Type notebook() for the GUI, and license() for information.        |
----------------------------------------------------------------------


sage: quadratic_residues(5)
```

```
[0, 1, 4]
sage: quadratic_residues(7)
[0, 1, 2, 4]
sage: MS = MatrixSpace(CyclotomicField(5),5,5)
sage: V = VectorSpace(CyclotomicField(5),5)
sage: v = V([0,1,-1,-1,1])
sage: z = CyclotomicField(5).gen()
sage: F5 = MS([[1,1,1,1,1],[1,z,z^2,z^3,z^4],[1,z^2,z^4,z^6,z^8],[1,z^3,z^6,z^9,z^(12)],[1,z^4,z^8,z^(12),z^(16)]])
sage: latex(F5)

\left(\begin{array}{rrrrr}
1&1&1&1&1\\
1&zeta5&zeta5^{2}&zeta5^{3}&-zeta5^{3} - zeta5^{2} - zeta5 - 1\\
1&zeta5^{2}&-zeta5^{3} - zeta5^{2} - zeta5 - 1&zeta5&zeta5^{3}\\
1&zeta5^{3}&zeta5&-zeta5^{3} - zeta5^{2} - zeta5 - 1&zeta5^{2}\\
1&-zeta5^{3} - zeta5^{2} - zeta5 - 1&zeta5^{3}&zeta5^{2}&zeta5
\end{array}\right)
sage: F5*v
(0, -2*zeta5^3 - 2*zeta5^2 - 1, 2*zeta5^3 + 2*zeta5^2 + 1, 2*zeta5^3 + 2*zeta5^2 + 1, -2*zeta5^3 - 2*zeta5^2 - 1)
sage: a = 2*z^3 + 2*z^2 + 1
sage: a^2
5
sage: exp(pi*I)
-1.00000000000000 + 0.000000000000000323829504877970*I
sage: zz=exp(2*pi*I/5)
sage: aa = 2*zz^3 + 2*zz^2 + 1
sage: aa
-2.23606797749979 + 0.000000000000000581756864903582*I
sage: zz=exp(-2*pi*I/5)
sage: aa = 2*zz^3 + 2*zz^2 + 1
sage: aa
-2.23606797749979 - 0.000000000000000588418203051332*I
sage: MS = MatrixSpace(CyclotomicField(5),5,5)
sage: z = CyclotomicField(5).gen()
sage: F5 = MS([[1,1,1,1,1],[1,z,z^2,z^3,z^4],[1,z^2,z^4,z^6,z^8],[1,z^3,z^6,z^9,z^(12)],[1,z^4,z^8,z^(12),z^(16)]])
sage: F5.fcp()
(x + -2*zeta5^3 - 2*zeta5^2 - 1) * (x + 2*zeta5^3 + 2*zeta5^2 + 1)^2 * (x^2 + 5)
sage: F5.fcp()
(x + -2*zeta5^3 - 2*zeta5^2 - 1) * (x + 2*zeta5^3 + 2*zeta5^2 + 1)^2 * (x^2 + 5)
sage: z^4
-zeta5^3 - zeta5^2 - zeta5 - 1
sage: MS.identity_matrix()

[1 0 0 0 0]
[0 1 0 0 0]
[0 0 1 0 0]
[0 0 0 1 0]
[0 0 0 0 1]
sage: A = F5 - (-2*z^3 - 2*z^2 - 1)*MS.identity_matrix()
sage: A.kernel()

Vector space of degree 5 and dimension 2 over Cyclotomic Field of order 5 and degree 4
Basis matrix:
[                    1                     0 -zeta5^3 - zeta5^2 - 1 -zeta5^3 - zeta5^2 - 1                      0]
[                    0                     1                     -1                     -1                      1]
sage: A.kernel().basis()

[
(1, 0, -zeta5^3 - zeta5^2 - 1, -zeta5^3 - zeta5^2 - 1, 0),
(0, 1, -1, -1, 1)
]
sage: -z^3 - z^2 - 1==z+z^4
True
sage: A = F5 + (-2*z^3 - 2*z^2 - 1)*MS.identity_matrix()
sage: A.kernel().basis()

[
(1, 1/2*zeta5^3 + 1/2*zeta5^2, 1/2*zeta5^3 + 1/2*zeta5^2, 1/2*zeta5^3 +
1/2*zeta5^2, 1/2*zeta5^3 + 1/2*zeta5^2)
]
sage: zz+zz^4
0.618033988749874 + 0.000000000000000115463194561016*I
sage: 4*(zz+zz^4)
2.47213595499949 + 0.000000000000000461852778244065*I
sage: (zz+zz^4)^2
0.381966011250079 + 0.000000000000000142720357376695*I
sage: (zz+zz^4)^4
```

```
0.145898033750295 + 0.000000000000000109028651262724*I
sage: (zz+zz^4)^5
0.0901699437494591 + 0.0000000000000000842292652849006*I
sage: (zz+zz^4+1)^2
2.61803398874982 + 0.000000000000000373646746498727*I
sage: 1/2*zz^3 + 1/2*zz^2
-0.809016994374949 - 0.000000000000000147104550762833*I
sage: MSI = MatrixSpace(CyclotomicField(20),5,5)
sage: z20 = CyclotomicField(20).gen()
sage: z5 = z20^4
sage: I in CyclotomicField(20)
False
sage: z4 = z20^5
sage: z4^4
1
sage: z4
zeta20^5
sage: z4^3
-zeta20^5
sage: F5I = MS([[1,1,1,1,1],[1,z5,z5^2,z5^3,z5^4],[1,z5^2,z5^4,z5^6,z5^8],[1,z5^3,z5^6,z5^9,z5^(12)],[1,z5^4,z5^8,z5^(12),z5^(16)]])
sage: A = F5I + z4*(-2*z5^3 - 2*z5^2 - 1)*MSI.identity_matrix()
sage: A.kernel().basis()

[
(0, 1, zeta20^7 + zeta20^6 - zeta20^5 - zeta20^4 + zeta20^3 - 2*zeta20 - 1,
      -zeta20^7 - zeta20^6 + zeta20^5 + zeta20^4 - zeta20^3 + 2*zeta20 + 1, -1)
]
sage: a = z20^7 + z20^6 - z20^5 - z20^4 + z20^3 - 2*z20 - 1
sage: a^5
165*zeta20^7 + 125*zeta20^6 - 105*zeta20^5 - 125*zeta20^4 + 45*zeta20^3 - 210*zeta20 - 193
sage: a in CyclotomicField(5)
False
sage: zz20=exp(-2*pi*I/20)
sage: zz20^5
0.000000000000000127675647831893 - 1.00000000000000*I
sage: A = F5I - z4*(-2*z5^3 - 2*z5^2 - 1)*MSI.identity_matrix()
sage: A.kernel().basis()

[
(0, 1, -zeta20^7 + zeta20^6 + zeta20^5 - zeta20^4 - zeta20^3 + 2*zeta20 - 1,
       zeta20^7 - zeta20^6 - zeta20^5 + zeta20^4 + zeta20^3 - 2*zeta20 + 1, -1)
]
sage: -zz^3 - zz^2 - 1
0.618033988749899 + 0.000000000000000294209101525666*I
sage: 1/2*zz^3 + 1/2*zz^2
-0.809016994374949 - 0.000000000000000147104550762833*I
sage: -zz20^7 + zz20^6 + zz20^5 - zz20^4 - zz20^3 + 2*zz20 - 1
0.284079043840412 + 0.000000000000000222044604925031*I
sage: zz20^7 + zz20^6 - zz20^5 - zz20^4 + zz20^3 - 2*zz20 - 1
-3.52014702134020 - 0.000000000000000222044604925031*I
sage: [j for j in range(5)]
[0, 1, 2, 3, 4]
sage: [j*2%5 for j in range(5)]
[0, 2, 4, 1, 3]
sage: [j*3%5 for j in range(5)]
[0, 3, 1, 4, 2]
sage: [j*4%5 for j in range(5)]
[0, 4, 3, 2, 1]
sage: F5.fcp()
(x + -2*zeta5^3 - 2*zeta5^2 - 1) * (x + 2*zeta5^3 + 2*zeta5^2 + 1)^2 * (x^2 + 5)
sage: -2*zz^3 - 2*zz^2 - 1
2.23606797749979 + 0.000000000000000588418203051332*I
sage:
```

The table below lists the multiplicity of the eigenvalues of F_N, for some small values of N:

N	mult. of $\sqrt{N}$	mult. of $-\sqrt{N}$	mult. of $i\sqrt{N}$	mult. of $-i\sqrt{N}$
4	2	1	0	1
5	2	1	1	1
6	2	2	1	1
7	2	2	1	2
8	3	2	1	2
9	3	2	2	2
10	3	3	2	2
11	3	3	2	3
12	4	3	2	3
13	4	3	3	3

In general, the multiplicity of $\lambda = \epsilon\sqrt{N}$ is equal to the rank of $M_{\lambda/\sqrt{N}}$ in (14). According to Good [G], this is

$\sqrt{N}$	$-\sqrt{N}$	$i\sqrt{N}$	$-i\sqrt{N}$
$[\frac{1}{4}(N+4)]$	$[\frac{1}{4}(N+2)]$	$[\frac{1}{4}(N-1)]$	$[\frac{1}{4}(N+1)]$

Here is the SAGE code verifying the last line of the first table above:

```
sage: p = 13
sage: MS = MatrixSpace(CyclotomicField(p),p,p)
sage: z = CyclotomicField(p).gen()
sage: zz = exp(-2*pi*I/p)
sage: r = lambda k: [z^(j*k) for j in range(p)]
sage: F = MS([r(k) for k in range(p)])
sage: F.fcp()
(x + -2*zeta13^11 - 2*zeta13^8 - 2*zeta13^7 - 2*zeta13^6 - 2*zeta13^5 - 2*zeta13^2 - 1)^3 *
  (x + 2*zeta13^11 + 2*zeta13^8 + 2*zeta13^7 + 2*zeta13^6 + 2*zeta13^5 + 2*zeta13^2 + 1)^4 * (x^2 + 13)^3
sage: 2*zz^11 + 2*zz^8 + 2*zz^7 + 2*zz^6 + 2*zz^5 + 2*zz^2 + 1
-3.60555127546399 - 0.00000000000000177635683940025*I
```

Here is an example of such a function. Let p be a prime number and let $\ell(j) = 1$ if j is a non-zero square mod p and -1 is j is a non-zero non-square mod p and $\ell(0) = 0$. This is called the *Legendre function* or the *quadratic residue symbol.* It turns out that (since the product of two squares is a square and the product of a square with a non-square is a non-square) that $\ell(jk) = \ell(j)\ell(k)$, for all non-zero j, k. Also, it can be checked that, for any $0 < k < p$, the set of multiples of k mod p is the same as the set of all integers mod p:

$$\{jk \mid j \in \mathbb{Z}/p\mathbb{Z}\}.$$

Therefore, the assumptions made in (15) and (16) above hold. The eigenvalues are sometimes called *Gauss sums* but we shall not discuss them further. Further information are available in the excellent papers by Good [G] and McClellan and Park [MP].

3.2 The DFT and the coefficients of the FS

We saw in (10) the approximation which motivated the definition of the DFT. If f is a "nice" function on $(0, P)$ and if $\vec{f} = (f(\frac{0}{N}P), f(\frac{1}{N}P), ..., f(\frac{N-1}{N}P))$ is the vector of sampled values of f then

$$FS(f)_k \approx \frac{1}{N} DFT(\vec{f})_k, \tag{17}$$

where k ranges over $\{0, 1, ..., N-1\}$. Based on the approximation in (10), one expects that the estimate (17) is only good when k/N is "small".

Example 25 *Let $f(x) = e^{-x}$ for $0 < x < 1$, extended periodically to $\mathbb{R}$ with period $P = 1$. The graph looks something like:*

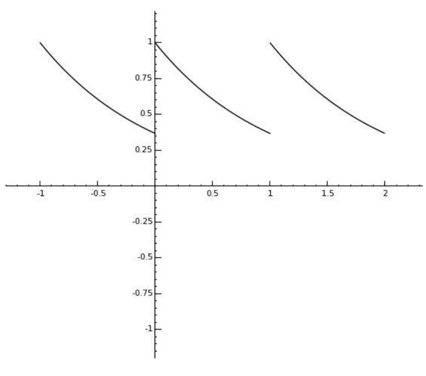


Figure 11: Graph of $f(x)$, $-1 < x < 2$.

We compute its k-th Fourier series coefficient:

$$c_k = \frac{1}{P}\int_0^P f(x)e^{-2\pi ikx/P}\,dx = \int_0^1 e^{-x}e^{-2\pi ikx}\,dx = \int_0^1 e^{-x-2\pi ikx}\,dx = \frac{1-e^{-1}}{1+2\pi ik},$$

for all $k \in \mathbb{Z}$. Now, let

$$\vec{f} = (f_0, f_1, ..., f_{N-1}) = (1, e^{-1/N}, e^{-2/N}, ..., e^{-(N-1)/N})$$

be the vector of sampled values. We compute its k-th DFT component:

$$\begin{aligned} DFT(\vec{f})_k &= \sum_{j=0}^{N-1} f_j e^{-2\pi ikj/N} \\ &= \sum_{j=0}^{N-1} e^{-j/N} e^{-2\pi ikj/N} \\ &= \sum_{j=0}^{N-1} (e^{(-1-2\pi ik)/N})^j \\ &= \frac{1-e^{-1-2\pi ik}}{1-e^{(-1-2\pi ik)/N}} \\ &= \frac{1-e^{-1}}{1-e^{(-1-2\pi ik)/N}}. \end{aligned}$$

The estimate (17) simply asserts in this case that

$$\frac{1}{N}\frac{1-e^{-1}}{1+2\pi ik} \approx \frac{1-e^{-1}}{1-e^{(-1-2\pi ik)/N}}.$$

Is this true?

If we use the approximation

$$e^{-x} = 1 - x + \frac{1}{2}x^2 + ...$$

we see that, for "small" $(-1-2\pi ik)/N$, $1-e^{(-1-2\pi ik)/N} \approx \frac{1}{N}(1+2\pi ik)$.

With $N = 10$, even the first few values aren't very close.

k	c_k	$DFT(\vec{f})_k$	$\lvert c_k - DFT(\vec{f})_k\rvert$
0	*0.63212*	*0.664253*	*0.03213*
1	*0.01561 - 0.09811i*	*0.04775 - 0.09478i*	*0.03231*
2	*0.003977 - 0.04998i*	*0.03615 - 0.04319i*	*0.03288*
3	*0.001774 - 0.03344i*	*0.03401 - 0.02287i*	*0.03392*

Here is the list plot of the values of $\lvert c_k - DFT(\vec{f})_k\rvert$

and here is the histogram plot:

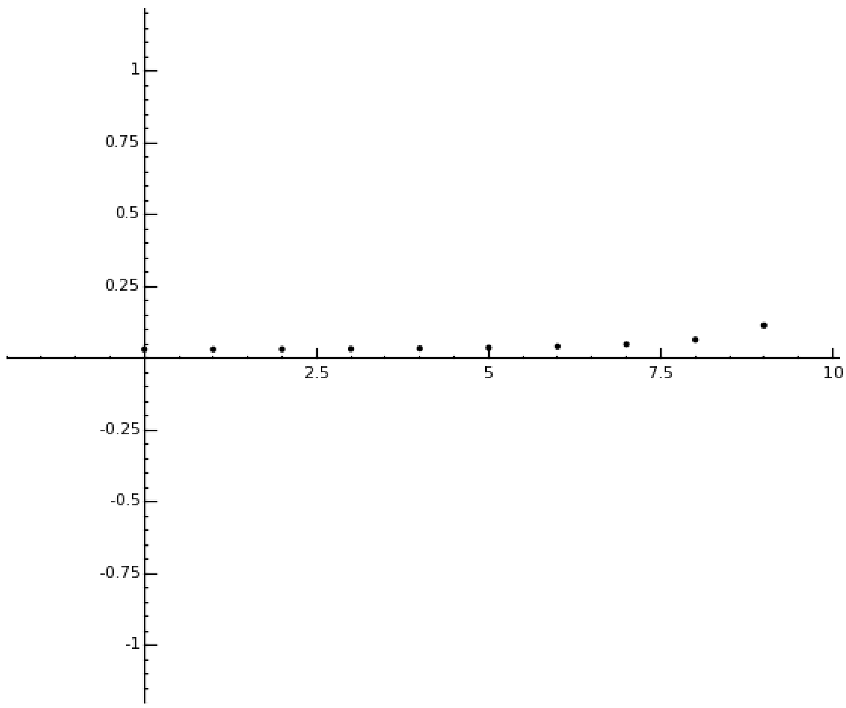


Figure 12: List plot of $|c_k - DFT(\vec{f})_k|$, $k = 0, ..., 9$.

You can see from these graphs that the larger k/N gets, the worse the approximation is.

When $N = 100$ the approximation is about 10 times better, as is to be expected.

k	c_k	$DFT(\vec{f})_k$	$\lvert c_k - DFT(\vec{f})_k\rvert$
0	*0.6321*	*0.6352*	*0.003173*
1	*0.01561 - 0.09812*I*	*0.01878 - 0.09808*I*	*0.003165*
2	*0.003977 - 0.04998*I*	*0.007143 - 0.04992*I*	*0.003166*
3	*0.001774 - 0.03344i*	*0.004939 - 0.03334*I*	*0.003167*

Here is the list plot of the values of $10|c_k - DFT(\vec{f})_k|$ (the error has been scaled up by a factor of 10 so that the plot comes out nicer):

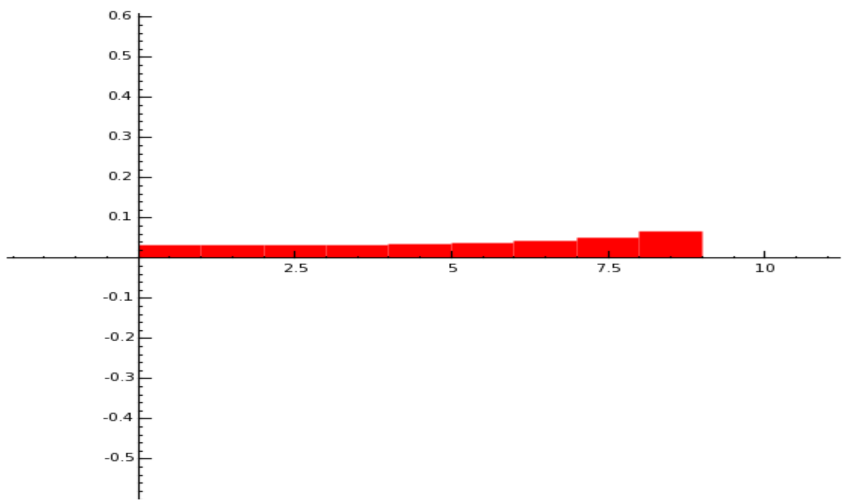


Figure 13: Histogram of $|c_k - DFT(\vec{f})_k|$, $k = 0, ..., 9$.

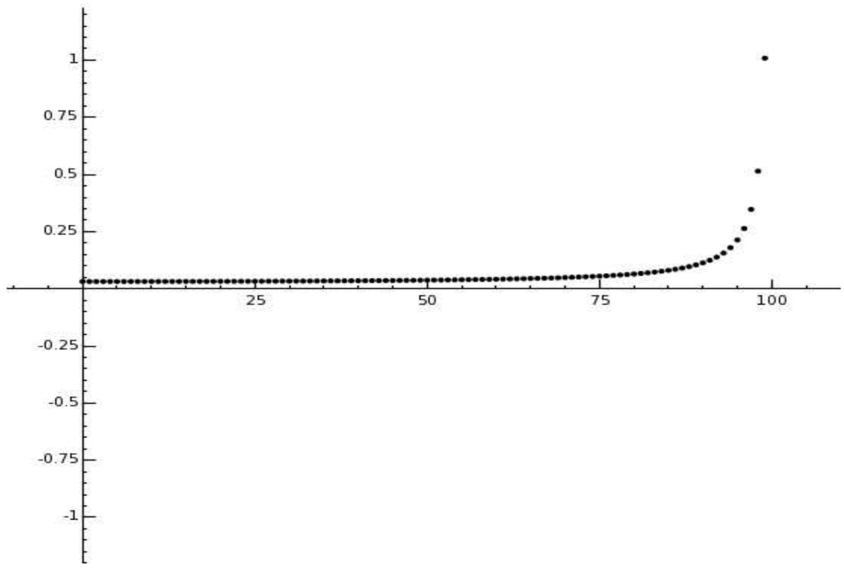


Figure 14: List plot of $|c_k - DFT(\vec{f})_k|$, $k = 0, ..., 99$.

Here is the `SAGE` *code for the $N = 10$ case:*

```
sage: CC5 = ComplexField(15)
sage: N = 10
```

```
sage: ck = lambda k:(1-exp(-1))/(1+2*pi*I*k)
sage: dftk = lambda k:(1/N)*(1-exp(-1))/(1-exp((-1-2*pi*I*k)/N))
sage: [CC5(ck(j)) for j in range(N)]

[0.6321,
 0.01561 - 0.09812*I,
 0.003977 - 0.04998*I,
 0.001774 - 0.03344*I,
 0.0009991 - 0.02511*I,
 0.0006397 - 0.02010*I,
 0.0004444 - 0.01675*I,
 0.0003265 - 0.01436*I,
 0.0002500 - 0.01257*I,
 0.0001976 - 0.01117*I]
sage: [CC5(dftk(j)) for j in range(N)]

[0.6642,
 0.04775 - 0.09479*I,
 0.03615 - 0.04319*I,
 0.03401 - 0.02287*I,
 0.03334 - 0.01024*I,
 0.03318 - 0.00000000000000005104*I,
 0.03334 + 0.01024*I,
 0.03401 + 0.02287*I,
 0.03615 + 0.04318*I,
 0.04775 + 0.09478*I]
sage: [abs(CC5(ck(j))-CC5(dftk(j))) for j in range(N)]

[0.03213,
 0.03231,
 0.03288,
 0.03392,
 0.03560,
 0.03825,
 0.04256,
 0.05021,
 0.06631,
 0.1161]
sage: L = [abs(CC5(ck(j))-CC5(dftk(j))) for j in range(N)]
sage: show(list_plot(L))
sage: J = range(N)
sage: s = IndexedSequence(L,J)
sage: (s.plot_histogram()).save("histogram-dftk-vs-ck10.png",xmin=-1,xmax=10,ymin=-0.5,ymax=0.5)
```

Let's try one more example.

Example 26 *Let $f(x) = x^2$ for $0 < x < 1$, extended periodically to $\mathbb{R}$ with period $P = 1$. The graph looks something like:*

You can enter this function and plot it's values, for $-1 < x < 2$, in `SAGE` *using the commands*

```
sage: f0 = (x+1)^2; f1 = x^2; f2 = (x-1)^2
sage: f = Piecewise([[(-1,0),f0],[(0,1),f1],[(1,2),f2]])
sage: show(f.plot())
```

We compute, for $k \neq 0$,

$$c_k = \int_0^1 x^2 e^{-2\pi ikx}\, dx = \frac{1}{2k^2\pi^2} + \frac{1}{2k\pi}i$$

The value for $k = 0$ is $c_0 = \frac{1}{3}$.

Here is the plot of the real part of the partial sum $\sum_{-10}^{10} c_k e^{2\pi ikx}$:

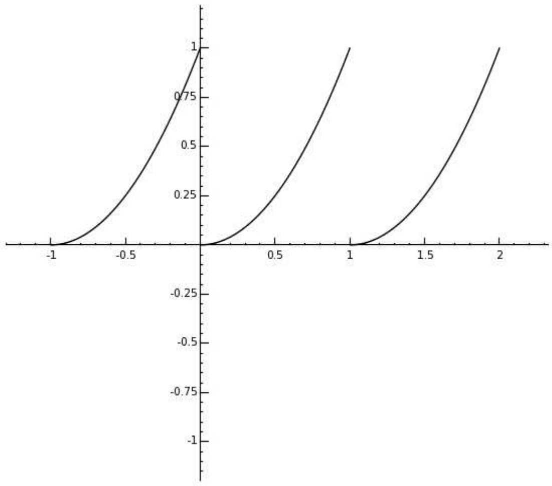


Figure 15: Graph of $f(x)$, $-1 < x < 2$.

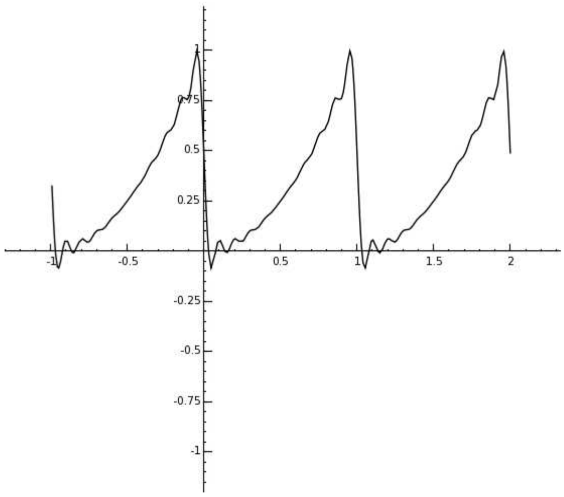


Figure 16: Graph of $S_{10}(x)$, $-1 < x < 2$.

```
sage: c = lambda k : I*((2*k^2*pi^2 - 1)/(4*k^3*pi^3) + 1/(4*k^3*pi^3)) + 1/(2*k^2*pi^2)
sage: ps = lambda x: 1/3+sum([(c(k)*exp(2*I*pi*k*x)).real() for k in range(-10,10) if k!=0])
sage: show(plot(ps,-1,2))
```

3.3 The DFT and convolutions

This section is based, not on Walker's book [W1] but on material in Frazier's well-written book [F].

Let $\mathbb{Z}/N\mathbb{Z}$ denote the abelian group of integers mod N, and let

$$V_N = \{f \mid f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{C}\}.$$

This is a $\mathbb{C}$-vector space which we can identify with the vector space $\mathbb{C}^N$ via the map $f \longmapsto (f(0), f(1), ..., f(N-1))$, $V_N \to \mathbb{C}^N$. You can visualize $\mathbb{Z}/N\mathbb{Z}$ as a circle with N equally spaced points and functions on $\mathbb{Z}/N\mathbb{Z}$ as "weights" on each of these points:

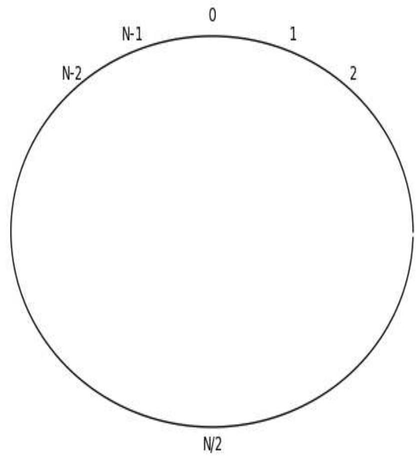


Figure 17: Circle representing $\mathbb{Z}/N\mathbb{Z}$.

Define *convolution* by

$$\begin{array}{ccc} V_N \times V_N & \to & V_N \\ (f,g) & \longmapsto & f*g, \end{array} \tag{18}$$

where

$$(f*g)(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(\ell)g(k-\ell).$$

In other words, if $\vec{f} = (f_0, f_1, ..., f_{N-1})$ and $\vec{g} = (g_0, g_1, ..., g_{N-1})$ then $\vec{f} * \vec{g}$ is another vector, whose k-th coordinate is given by

$$(\vec{f} * \vec{g})_k = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f_\ell g_{k-\ell},$$

where are subscripts are computed mod N and represented in the set $\{0, 1, ..., N-1\}$ This binary operation on V_N is commutative. In other words, $f*g = g*f$: if $\ell' = k - \ell$ then

$$(f*g)(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(\ell)g(k-\ell) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(k-\ell')g(\ell').$$

We will be done once we show that (for each $k \in \mathbb{Z}/N\mathbb{Z}$) as ℓ ranges over all of $\mathbb{Z}/N\mathbb{Z}$, so does $\ell' = k - \ell$. But if ℓ' misses something in $\mathbb{Z}/N\mathbb{Z}$, say x, then $k - \ell \neq x$ for all $\ell \in \mathbb{Z}/N\mathbb{Z}$, and so $\ell \neq k - x$. This is a contradiction, so $\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(\ell)g(k-\ell) = \sum_{\ell' \in \mathbb{Z}/N\mathbb{Z}} f(k-\ell')g(\ell') = (g*f)(k)$, as desired.

Example 27 Here is a `SAGE` session for computing the convolution:

```
sage: F = CyclotomicField(16)
sage: J = range(16)
sage: A = [F(0) for i in J]; A[0] = F(1); A[1] = F(1); A[2] = F(1); A[3] = F(1)
sage: z = F.gen()
sage: B = [z^(i) for i in J]
sage: sA = IndexedSequence(A,J)
sage: sB = IndexedSequence(B,J)
sage: sA.convolution(sB)

[1,
 zeta16 + 1,
 zeta16^2 + zeta16 + 1,
 zeta16^3 + zeta16^2 + zeta16 + 1,
 zeta16^4 + zeta16^3 + zeta16^2 + zeta16,
 zeta16^5 + zeta16^4 + zeta16^3 + zeta16^2,
 zeta16^6 + zeta16^5 + zeta16^4 + zeta16^3,
 zeta16^7 + zeta16^6 + zeta16^5 + zeta16^4,
 zeta16^7 + zeta16^6 + zeta16^5 - 1,
 zeta16^7 + zeta16^6 - zeta16 - 1,
 zeta16^7 - zeta16^2 - zeta16 - 1,
 -zeta16^3 - zeta16^2 - zeta16 - 1,
 -zeta16^4 - zeta16^3 - zeta16^2 - zeta16,
 -zeta16^5 - zeta16^4 - zeta16^3 - zeta16^2,
 -zeta16^6 - zeta16^5 - zeta16^4 - zeta16^3,
 -zeta16^7 - zeta16^6 - zeta16^5 - zeta16^4,
 -zeta16^7 - zeta16^6 - zeta16^5,
 -zeta16^7 - zeta16^6,
 -zeta16^7,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Let $\zeta = \zeta_N$ denote a primitive N^{th} root of unity in F ($\overline{W}$, in the notation above, is one good choice). Recall, for $g \in V_N$, the *discrete Fourier transform* of g was defined by

$$g^\wedge(\lambda) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(\ell)\zeta^{\ell\lambda}, \qquad \lambda \in \mathbb{Z}/N\mathbb{Z}.$$

Also, we defined the *inverse discrete Fourier transform* of G by

$$G^\vee(\ell) = \frac{1}{N} \sum_{\lambda \in \mathbb{Z}/N\mathbb{Z}} G(\lambda)\zeta^{-\ell\lambda}, \qquad \ell \in \mathbb{Z}/N\mathbb{Z}.$$

A basic and very useful fact about the Fourier transform is that *the Fourier transform of a convolution is the product of the Fourier transforms.* Here's the proof:

$$\begin{array}{ll} (f*g)^\wedge(\lambda) & = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} \sum_{k \in \mathbb{Z}/N\mathbb{Z}} f(k)g(\ell-k)\zeta^{\ell\lambda} \\ & = \sum_{k \in \mathbb{Z}/N\mathbb{Z}} f(k) \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(\ell-k)\zeta^{\ell\lambda} \\ & = \sum_{k \in \mathbb{Z}/N\mathbb{Z}} f(k)\zeta^{k\lambda} \sum_{\ell' \in \mathbb{Z}/N\mathbb{Z}} g(\ell')\zeta^{\ell'\lambda} \\ & = f^\wedge(\lambda)g^\wedge(\lambda). \end{array}$$

In coordinate notation:

$$DFT(\vec{f} * \vec{g})_k = DFT(\vec{f})_k DFT(\vec{g})_k,$$

for all $0 \le k \le N-1$.

Definition 28 For $h \in V_N$, define $M_h : V_N \to V_N$ by

$$M_h(f) = (hf^\wedge)^\vee.$$

If $\times$ denotes the componentwise product of two vectors,

$$(a_0, a_1, ..., a_{N-1}) \times (b_0, b_1, ..., b_{N-1}) = (a_0b_0, a_1b_1, ..., a_{N-1}b_{N-1}),$$

then, in coordinate notation,

$$M_{\vec{h}}(\vec{f}) = DFT^{-1}(\vec{h} \times DFT(\vec{f})),$$

for $\vec{f}, \vec{h} \in \mathbb{C}^N$. A linear transformation $T : V_N \to V_N$ of the form $T = M_h$, for some $h \in V_N$, is called a *Fourier multiplier operator*.

In other words, a Fourier multiplier operator (represented in the standard basis) is a linear transformation of the form $F_N^{-1} D F_N$, where D is an $N \times N$ diagonal matrix. Note that the product of two Fourier multiplier operators is a Fourier multiplier operator: $M_{h_1} M_{h_2} = M_{h_1 h_2}$.

Stereo systems and FMOs: Here is one way in which Fourier multiplier operators can be thought of in terms of Dolby stereo (this is a grossly over-simplied description but you will get the idea). Dolby cuts off the high frequencies, often which are crackles and pops and other noise in the channel, making the music sound nicer. How does it do that? If $f \in V_N$ represents the digital sample of the sound, $DFT(f) = f^\wedge$ represents the frequencies of the sound. To cut off the high frequencies, multiply $DFT(f)$ by some ("low-pass filter") $h \in V_N$ which is 0 on the high frequencies and 1 on the rest: you get $hf^\wedge$. To recover the sound from this, take the inverse DFT, $(hf^\wedge)^\vee$. This is the same sound, but without the high frequencies. This "filtered sound" is an example of a Fourier multiplier operator.

The following result appears as Theorem 2.19 of [F]. It characterizes the Fourier multiplier operators.

Theorem 29 *Let $T : V_N \to V_N$ denote a linear operator. The following are equivalent:*

1. *T is translation invariant.*
2. *The matrix A representing T in the standard basis is circulant.*
3. *T is a convolution operator.*
4. *T is a Fourier multiplier operator.*
5. *The matrix B representing T in the Fourier basis is diagonal.*

We shall define all these terms (convolution operator, etc) give some examples, and prove this theorem.

Remark 1 As a consequence of the proof below, we shall show that, for all $f, g \in V_N$,

$$f * g = (g^\wedge f^\wedge)^\vee.$$

Let $M(N)$ denote the number of multiplications required to compute the DFT on V_N. The above identity implies that the number of multiplications required to compute the convolution $f * g$ is at most $2 \cdot M(N) + 1$. We shall see in §6.1 that $M(N) \leq N(\log_2(N) + 2)$. (This remark shall be applied in §6.2 below.)

Definition 30

- Let $\tau : V_N \to V_N$ denote the *translation map*: $(\tau f)(x) = f(x+1)$ (addition in $\mathbb{Z}/N\mathbb{Z}$). Note

$$\tau : (x_0, x_1, ..., x_{n-1}) \longmapsto (x_1, x_2, ..., x_{N-1}, x_0).$$

- We say an operator T is *translation invariant* if the diagram

$$\begin{tikzcd} V_N \arrow[r, "\tau"] \arrow[d, "T"'] & V_N \arrow[d, "T"'] \\ V_N \arrow[r, "\tau"] & V_N \end{tikzcd}$$

commutes. (This phrase will be explained below.)

- Define the *convolution operator*[6] *associated to* g,

$$T_g : V_N \to V_N,$$

by $T_g(f) = f * g$.

Remark 2 Here is a remark on the grammar used in the diagramatical definition of translation invariance above. The phrase "diagram commutes" is a fancy way to say that, for each $f \in V_N$ (picking an element in the copy of V_N in the upper left hand corner), the element $T(\tau(f)) \in V_N$ (mapping from the upper left corner along the top arrow and down the right arrow $\tau(f) \longmapsto T(\tau(f))$) is equal to the element $\tau(T(f))$ (mapping down the left arrow and along the bottom arrow), as functions on $\mathbb{Z}/N\mathbb{Z}$. In other words, *T is translation invariant if and only if, for all $k \in \mathbb{Z}/N\mathbb{Z}$ and all $f \in V_N$, we have $T(\tau(f))(k) = \tau(T(f))(k)$.*

Example 31 Let's look again at the operator neg : $V_N \to V_N$, which sends $f(k)$ to $f(-k)$. Is this translation invariant? To answer this, we must see whether or not $(\tau(\text{neg}f))(k) = (\text{neg}(\tau f))(k)$, for each $k \in \mathbb{Z}/N\mathbb{Z}$ and each $f \in V_N$. We have, for example, $(\tau(\text{neg}f))(0) = (\text{neg}f))(1) = f(-1) = f(N-1)$ and $(\text{neg}(\tau f))(0) = (\tau f)(-0) = (\tau f)(0) = f(1)$. In general, these two coordinares are different, so neg is not translation invariant[7].

[6]As usual, the term "operator" is reserved for a linear transformation from a vector space to *itself*.

[7] However, if you restrict `neg` to the subspace of V_N of functions f for which $f(N-\ell) = f(\ell)$ then it is translation invariant (in fact, on this subspace, it is the identity).

Lemma 32 *The convolution operator T_g is translation invariant. In other words, the diagram*

$$\begin{array}{ccc} V_N & \xrightarrow{\tau} & V_N \\ {\scriptstyle T_g}\downarrow & & {\scriptstyle T_g}\downarrow \\ V_N & \xrightarrow{\tau} & V_N \end{array}$$

commutes, for all $g \in V_N$.

In terms of the above theorem, this lemma says "3 $\implies$ 1".

proof: Recall

$$T_g(f)(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(\ell)g(k-\ell),$$

for $k \in \mathbb{Z}/N\mathbb{Z}$. We have

$$\begin{array}{ll} T_g(\tau(f))(k) & = (\tau(f) * g)(k) \\ & = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} (\tau f)(\ell) g(k-\ell) \\ & = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} f(\ell+1) g(k-\ell) \\ & = \sum_{\ell' \in \mathbb{Z}/N\mathbb{Z}} f(\ell') g(k+1-\ell') \quad (\ell' = \ell+1) \\ & = T_g(f)(k+1) = \tau(T_g(f))(k). \end{array}$$

$\square$

Definition 33 An $N \times N$ matrix A is *circulant* if and only if

$$A_{k,\ell} = A_{k+1\,(\mathrm{mod}\ N),\ \ell+1\,(\mathrm{mod}\ N)},$$

for all $0 \leq k \leq N-1$, $0 \leq \ell \leq N-1$.

In other words, to go from one row to the next in a circulant matrix, just "rotate" or cyclically permute the rows. In particular, (setting $k = \ell = 1$) $A_{1,1} = A_{2,2} = ... = A_{N,N}$, so all the diagonal entries must be the same.

Example 34 *Let $N = 5$. The matrix*

$$C = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

is circulant.

Lemma 35 *A linear transformation $T : V_N \to V_N$ is translation invariant if and only if the matrix representing it in the standard basis is circulant.*

In terms of the above theorem, this lemma says "1 $\iff$ 2".

proof: Since $T : V_N \to V_N$ is linear, with respect to the standard vasis it is represented by an $N \times N$ matrix

$$Tf = A\vec{f}, \qquad A = (A_{i,j}),$$

where $\vec{f} = {}^t(f(0), f(1), ..., f(N-1))$. In other words, $T(f)(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{k,\ell} f(\ell)$, for $k \in \mathbb{Z}/N\mathbb{Z}$. For $k \in \mathbb{Z}/N\mathbb{Z}$, we have

$$\begin{aligned} T(\tau(f))(k) &= \textstyle\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{k,\ell} (\tau f)(\ell) \\ &= \textstyle\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{k,\ell} f(\ell + 1) \\ &= \textstyle\sum_{\ell' \in \mathbb{Z}/N\mathbb{Z}} A_{k,\ell'-1} f(\ell'), \end{aligned}$$

where the 2^{nd} subscript of $A_{i,j}$ is taken mod N. Changing the dummy variable ℓ' to ℓ, this becomes

$$T(\tau(f))(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{k,\ell-1} f(\ell).$$

On the other hand,

$$\tau(T(f))(k) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{k+1,\ell} f(\ell).$$

Comparing coefficients, it follows that the linear transformation T is translation invariant if and only if its matrix A satisfies: $A_{k,\ell} = A_{k+1 \,(\mathrm{mod}\ N),\ \ell+1 \,(\mathrm{mod}\ N)}$, for all $0 \le k \le N - 1$, $0 \le \ell \le N - 1$. But this is true if and only if A is circulant. $\square$

Example 36 Let $N = 5$ and let $T : V_5 \to V_5$ be defined by

$$T(f)(k) = (C\vec{f})_k,$$

where C is as in Example 34 and $\vec{f}$ is the column vector $\vec{f} = (f(0), f(1), f(2), f(3), f(4))$, for $f \in V_5$. Here is an example:

k	0	1	2	3	4
$f(k)$	1	0	1	0	0
$\tau f(k)$	0	1	0	1	0
$Tf(k)$	4	7	5	8	6
$\tau(Tf)(k)$	6	4	7	5	8
$T(\tau f)(k)$	6	4	7	5	8

We see in this example that $T\tau(f) = \tau T(f)$, consistent with the fact "(2) $\implies$ (1)".

Lemma 37 *If a linear transformation $T : V_N \to V_N$ is translation invariant then it is a convolution map.*

In terms of the above theorem, this lemma says "1 $\implies$ 3".

proof: Let T denote a translation invariant linear operator. Define $g \in V_N$ by $g(k) = A_{0,-k \pmod N}$, $k \in \mathbb{Z}/N\mathbb{Z}$. Then

$$T(f)(0) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} A_{0,\ell} f(\ell) = \sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(-\ell) f(\ell),$$

for all $f \in V_N$. Replacing f by a translation $\pmod N$ (note g is periodic with period N) gives

$$\begin{aligned} T(f)(k) &= \tau^k(T(f))(0) = T(\tau^k f)(0) \\ &= \textstyle\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(-\ell)\tau^k f(\ell) \\ &= \textstyle\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(-\ell) f(\ell + k) \\ &= \textstyle\sum_{\ell' \in \mathbb{Z}/N\mathbb{Z}} g(k - \ell') f(\ell') \quad (\ell' = \ell + k) \\ &= \textstyle\sum_{\ell \in \mathbb{Z}/N\mathbb{Z}} g(k - \ell) f(\ell), \end{aligned}$$

for all $f \in V_N$. In other words, T is a convolution map. $\square$

Using the above lemmas, we see the connection between maps given by circulant matrices and convolution operators.

Lemma 38 *If $T : V_N \to V_N$ is a convolution operator then, represented in the Fourier basis, T is diagonal.*

This says (3) $\implies$ (5).

proof: By hypothesis, $T = T_g$, for some $g \in V_N$. Let us compute the action of T_g on the function $b_k \in V_N$, where $b_k(\ell) = e^{2\pi i k\ell/N}$. We have

$$\begin{array}{ll} T_g(b_j)(k) & = \sum_{\ell\in\mathbb{Z}/N\mathbb{Z}} e^{2\pi i j\ell/N} g(k-\ell) \\ & = \sum_{\ell'\in\mathbb{Z}/N\mathbb{Z}} e^{2\pi i j(k-\ell')/N} g(\ell') \\ & = e^{2\pi i jk/N} g^\wedge(j) = g^\wedge(j) b_j(k), \end{array}$$

so $T_g(b_j) = g^\wedge b_j$. If T is represented by $A = (A_{j,k})$, in the Fourier basis, then

$$T(b_j) = \sum_j A_{j,k} b_k,$$

for $0 \leq j \leq N-1$. Since the $\{b_j\}_{j=0}^{N-1}$ are linearly independent (they are a basis, after all!), we may compare coefficients to conclude: $A_{j,j} = g^\wedge(j)$ for all j and $A_{j,k} = 0$ if $j \neq k$. In other words, A is diagonal, as desired. $\Box$

In terms of the above theorem, this computation proves "$3 \implies 5$".

Lemma 39 *$T : V_N \to V_N$ is a convolution operator if and only if T is a Fourier multiplier operator.*

This says (3) $\iff$ (4).

proof: Suppose $T : V_N \to V_N$ is a convolution operator, say $T = T_g$, for some $g \in V_N$. Let $h = g^\wedge$ denote the inverse discrete Fourier transform of g. The claim is that $T_g = M_h$, where M_h is defined as in Definition 28. Since the Fourier transform of the convolution is the product of the Fourier transforms, for each $f \in V_N$, we have $(f*g)^\wedge = f^\wedge g^\wedge$. Taking inverse Fourier transforms of both sides gives

$$T_g(f) = f * g = (g^\wedge f^\wedge)^\vee = M_{g^\wedge}(f).$$

Therefore, if T is a convolution operator then it is also a Fourier multiplier operator.

Conversely, suppose T is a Fourier multiplier operator, say $T = M_h$, for some $h \in V_N$. Let $g = h^\vee$. The claim is that $T_g = M_h$. The proof of this case also follows from the identity $(f*g)^\wedge = f^\wedge g^\wedge$. $\Box$

Lemma 40 *If $T : V_N \to V_N$ is diagonal, represented in the Fourier basis, then T is translation invariant.*

This says (5) $\implies$ (1).

proof: By hypothesis, there are constants $\gamma_i \in \mathbb{C}$ such that $T(b_\ell)(k) = \gamma_\ell b_\ell(k)$, for each ℓ with $0 \leq \ell \leq N-1$ and each $k \in \mathbb{Z}/N\mathbb{Z}$. Therefore, if $f \in V_N$ is expressed in terms of the Fourier basis as $f(x) = \sum_\ell c_\ell b_\ell(x)$ (for $c_\ell \in \mathbb{C}$) then

$$T(f)(k) = \sum_\ell c_\ell \gamma_\ell b_\ell(k), \qquad k \in \mathbb{Z}/N\mathbb{Z}.$$

Now we compute

$$\tau(Tf)(k) = Tf(k+1) = \sum_\ell c_\ell \gamma_\ell b_\ell(k+1) = \sum_\ell e^{-2\pi i\ell/N} c_\ell \gamma_\ell b_\ell(k),$$

and

$$\tau f(k) = \sum_\ell c_\ell \tau b_\ell(k) = \sum_\ell c_\ell b_\ell(k+1) = \sum_\ell e^{-2\pi i\ell/N} c_\ell b_\ell(k),$$

so

$$T(\tau f)(k) == \sum_\ell e^{-2\pi i\ell/N} c_\ell T(b_\ell)(k) = \sum_\ell e^{-2\pi i\ell/N} c_\ell \gamma_\ell b_\ell(k).$$

This implies $\tau T(f) = T\tau(f)$, for all $f \in V_N$, as desired. $\Box$

We have: (3) $\implies$ (1) (Lemma 32), (1) $\iff$ (2) (Lemma 35), (1) $\implies$ (3) (Lemma 37), (3) $\implies$ (5) (Lemma 38), (3) $\iff$ (4) (Lemma 39), and (5) $\implies$ (1) (Lemma 40). These lemmas taken together finished the proof of the theorem.

In the next section, we shall give an example of how the Cesàro filter gives rise to a Fourier multiplier operator.

4 Filters and reconstruction

Here is a problem which is sometimes called the *reconstruction problem*: Suppose we know the Fourier series coefficients c_n of $f(x)$ but we don't know

$f(x)$ itself. How do we "reconstruct" $f(x)$ from its FS coefficients? The theoretical answer to this is very easy, it is (by Dirichlet's Theorem 8) simply the FS expansion:

$$f(x) = \sum_{n \in \mathbb{Z}} c_n e^{2\pi i n x/P}.$$

Suppose that we modify this problem into a more practical one: Suppose we know the Fourier series coefficients c_n of $f(x)$ but we don't know $f(x)$ itself. Given some error tolerance $\delta > 0$, how do we "reconstruct" (efficiently) $f(x)$, with error at most δ, from its FS coefficients? Practically speaking, a "filter" is a sequence of weights you apply to the FS coefficients to accomplish some aim (such as approximating $f(x)$ "quickly", or filtering out "noise", or ...). Can we weight the terms in the partial sums of the FS to compute an approximation to the value of a FS in a more efficient way than simply looking at partial sums alone? In this section, we examine several filters and see that the answer to this question is, in a specific sense, "yes".

4.1 Dirichlet's kernel

Let

$$S_M(x) = \sum_{j=-M}^{M} c_k e^{2\pi i j x/P}$$

denote the M-th partial sum of the FS of f. (This is a filter, where the weights are all 1 from $-M$ to M and 0 elsewhere. Sometimes the term "rectangular window" is seen in the literature for this.) Here is an integral representation for S_M. First, recall that

$$c_j = \frac{1}{P} \int_0^P f(t) e^{-2\pi i j t/P}\, dt,$$

so

$$\begin{array}{rl} S_M(x) & = \sum_{j=-M}^{M} \frac{1}{P} (\int_0^P f(t) e^{-2\pi i j t/P}\, dt) e^{2\pi i j x/P} \\ & = \frac{1}{P} \int_0^P f(t) \sum_{j=-M}^{M} e^{2\pi i j (x-t)/P}\, dt \\ & = \int_0^P f(t) K_M^D(x-t)\, dt \\ & = f * K_M^D(x), \end{array} \tag{19}$$

where $K_M^D(z) = \frac{1}{P}\sum_{j=-M}^{M} e^{2\pi ijz/P}$. (Note that Walker defined the convolution in a slightly different way than we do.) This function is called the *Dirichlet kernel* . It turns out this function has a nice closed-form expression:

$$K_M^D(x) = \frac{\sin((2M+1)\frac{x\pi}{P})}{P\sin(\frac{x\pi}{P})}. \tag{20}$$

Some plots of this, for various values of M, are given below. You can see that the graphs get 'spikier and spikier" (approaching the Dirac delta function, δ) as M gets larger and larger.

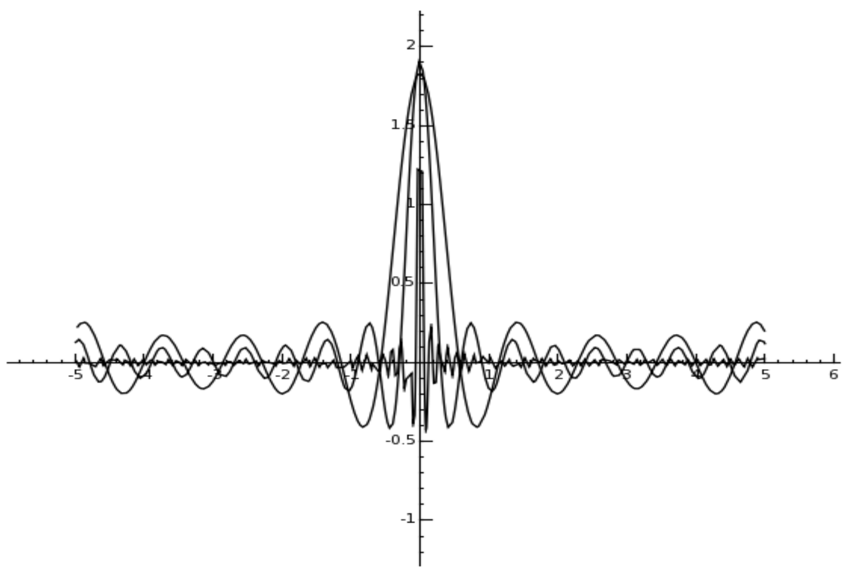


Figure 18: List plot of values of $K_M^D(x)$, $M = 5, 10, 50$, $P = 2\pi$.

The `SAGE` commands for this are as follows:

```
sage: M = 5
sage: f = lambda z: (1/(M+1))*(sin((2*M+1)*z/2)/sin(z/2))
sage: P1 = plot(f,-5,5)
sage: M = 10
sage: f = lambda z: (1/(M+1))*(sin((2*M+1)*z/2)/sin(z/2))
sage: P2 = plot(f,-5,5)
sage: M = 50
sage: f = lambda z: (1/(M+1))*(sin((2*M+1)*z/2)/sin(z/2))
```

```
sage: P3 = plot(f,-5,5)
sage: show(P1+P2+P3)
```

Here is the proof of (20):

$$\begin{aligned}\sum_{j=-M}^{M} e^{2\pi ijz/P} &= e^{-2\pi iMz/P}\sum_{j=0}^{2M} e^{2\pi ijz/P}\\ &= e^{-2\pi iMz/P}\frac{e^{2\pi i(2M+1)z/P}-1}{e^{2\pi iz/P}-1}\\ &= e^{-2\pi iMz/P}\frac{(e^{\pi i(2M+1)z/P}-e^{-\pi i(2M+1)z/P})/2i}{(e^{\pi iz/P}-e^{-\pi iz/P})/2i}\\ &= e^{-2\pi iMz/P+\pi i(2M+1)z/P-\pi iz/P}\frac{\sin(\pi(2M+1)z/P)}{\sin(\pi z/P)}\\ &= \frac{\sin(\pi(2M+1)z/P)}{\sin(\pi z/P)},\end{aligned}$$

as desired.

4.2 Cesàro filters

Aside: Here is a historical/mathematical remark explaining the idea behind this. Start with any infinite series,

$$\sum_j a_j = a_0 + a_1 + a_2 + a_3 + \dots\ ,$$

which, for the sake of this discussion, let's assume converges absolutely. This means that the series of partial sums

$$s_0 = a_0,\ s_1 = a_0 + a_1,\ , s_2 = a_0 + a_1 + a_2, \dots$$

has a limit - namely the value of the series $\sum_j a_j$. In particular, the series of arithmetic means

$$\begin{gathered} m_0 = s_0 = a_0,\ m_1 = \frac{s_0+s_1}{2} = \frac{1}{2}(a_0 + a_0 + a_1) = a_0 + \frac{1}{2}a_1,\\ m_2 = \frac{s_0+s_1+s_2}{3} = \frac{1}{3}(a_0 + a_0 + a_1 + a_0 + a_1 + a_2) = a_0 + \frac{2}{3}a_1 + \frac{1}{3}a_2, \dots\end{gathered}$$

also has a limiting value, namely the value of the series $\sum_j a_j$. (After all, you are simply averaging values which themselves have a limiting value.) In general,

$$m_J = \sum_{j=1}^{J}(1 - \frac{j}{J+1})a_j.$$

In the early 1900's the basic observation that the limit of the m_J's, as $J \to \infty$, is equal to $\sum_j a_j$ was applied by Fejér to the study of Fourier series. A more general convergence test was devised by Cesàro, who generalized the "weights" $1 - \frac{j}{J+1}$ used above. For historical reasons, the "weights" $1 - \frac{j}{J+1}$ are sometimes called *Cesàro filters*. □

As usual, let $S_M(x) = \sum_{j=-M}^{M} c_k e^{2\pi ijx/P}$ denote the M-th partial sum of the FS of f and let

$$S_M^C(x) = \sum_{j=-M}^{M} (1 - \frac{|j|}{M}) c_j e^{2\pi ijx/P} = \frac{1}{M} \sum_{j=0}^{M} S_j(x).$$

The above historical discussion motivates the following terminology.

Definition 41 The *M-th Cesàro-filtered* partial sum of the FS of f is the function $S_M^C = S_{M,f}^C$ above.

As the graphs show, as M gets larger, the S_M^C's approximate f "more smoothly" than the S_M's do. The factor $(1-\frac{|j|}{M})$ have the effect of "smoothing out" the "Gibbs phenomenon spikes" you see near the jump discontinuities.

Example 42 *As an example, here are the plots of some partial sums of the Fourier series, and filtered partial sums of the Fourier series.*

Let $f(x) = e^{-x}$, $0 < x < 1$ as in Example 25 above.

We shall use list plots, since they are easy to construct in `SAGE` *. Here is f again, but as a list plot:*

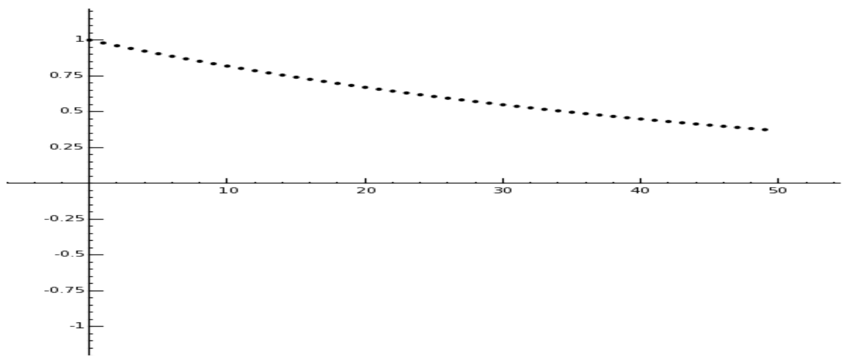


Figure 19: List plot of values of e^{-x}, $0 < x < 1$.

S_5:

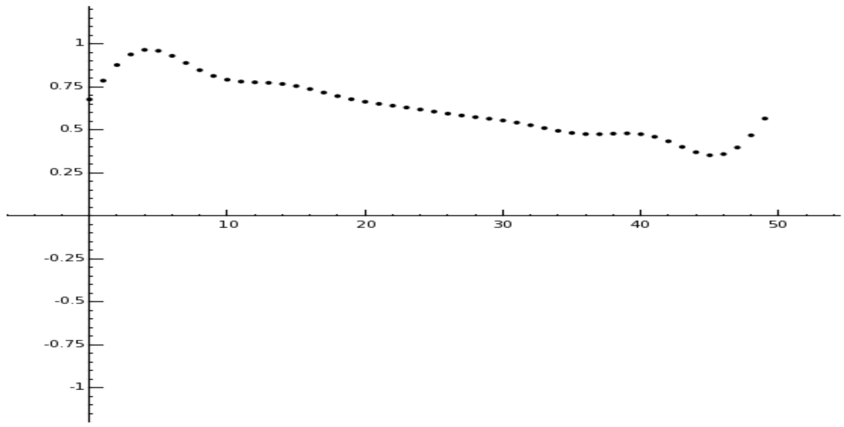


Figure 20: List plot of values of $S_5(x)$, $0 < x < 1$.

S_5^C:

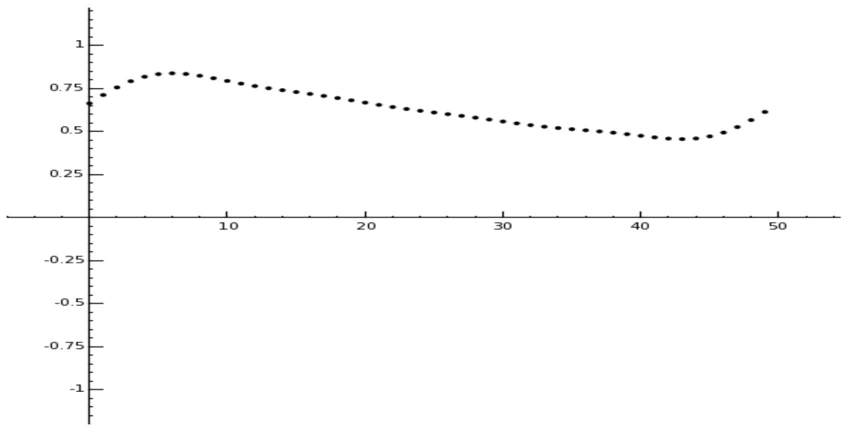


Figure 21: List plot of values of $S_5^C(x)$, $0 < x < 1$.

Note how the Gibbs phenomenom of S_5 is "smoothed out" by the filter - the graph of S_5^C seems to be less "bumpy".

S_{10}:

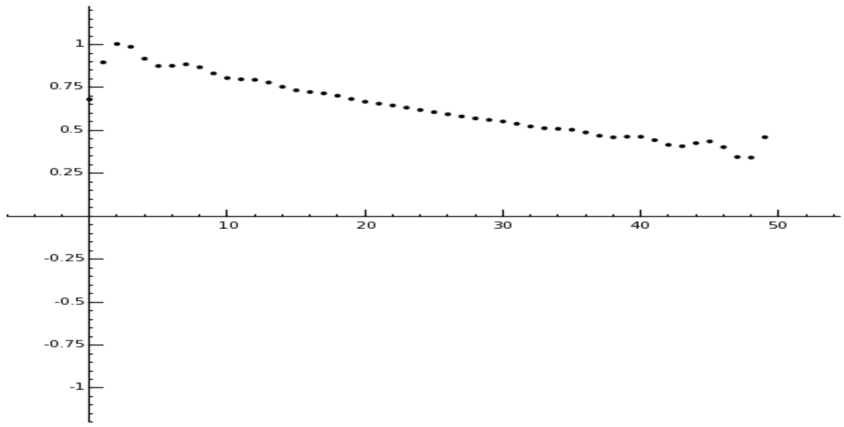


Figure 22: List plot of values of $S_{10}(x)$, $0 < x < 1$.

S_{10}^{C}:

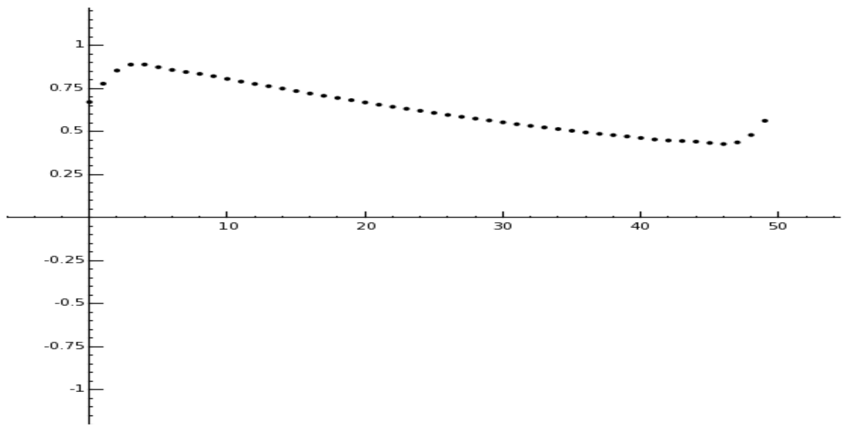


Figure 23: List plot of values of $S_{10}^{C}(x)$, $0 < x < 1$.

S_{25}:

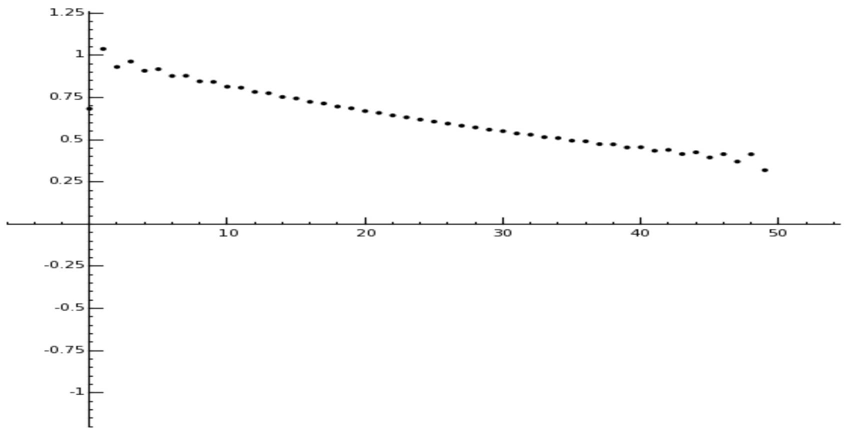


Figure 24: List plot of values of $S_{25}(x)$, $0 < x < 1$.

S_{25}^C:

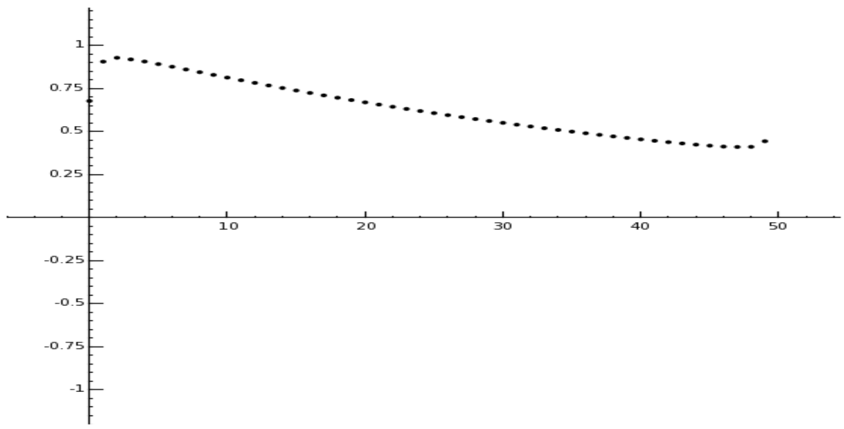


Figure 25: List plot of values of $S_{25}^C(x)$, $0 < x < 1$.

In each case, we see that the Cesàro filter "smooths out" the the Gibbs phenomenom of the partial sums of the Fourier series.

Here is the `SAGE` *code for the* $M = 25$ *case:*

```
sage: FSf = lambda x:(sum([ck(j)*exp(2*pi*I*j*x) for j in range(-25,25)])).real()
sage: L = [FSf(j/50) for j in range(50)]
sage: show(list_plot(L))

sage: FSf = lambda x:(sum([(1-abs(j)/25)*ck(j)*exp(2*pi*I*j*x) for j in range(-25,25)])).real()
sage: L = [FSf(j/50) for j in range(50)]
sage: show(list_plot(L))
```

Here is an integral representation for S_M^C. First, recall that

$$c_j = \frac{1}{P}\int_0^P f(t)e^{-2\pi ijt/P}\,dt,$$

so

$$\begin{aligned} S_M^C(x) &= \textstyle\sum_{j=-M}^M (1-\frac{|j|}{M})\frac{1}{P}(\int_0^P f(t)e^{-2\pi ijt/P}\,dt)e^{2\pi ijx/P} \\ &= \textstyle\frac{1}{P}\int_0^P f(t)\sum_{j=-M}^M (1-\frac{|j|}{M})e^{2\pi ij(x-t)/P}\,dt \\ &= \textstyle\int_0^P f(t)K_M(x-t)\,dt, \end{aligned} \tag{21}$$

where

$$K_M(z) = \frac{1}{P}\sum_{j=-M}^{M}(1-\frac{|j|}{M})e^{2\pi ijz/P}.$$

This function is called the *Fejér kernel* (it is also sometimes referred to as the "point spread function" of the filter). This has a simpler expression,

$$K_M(z) = \frac{1}{P}\frac{1}{M+1}(\frac{\sin((M+1)\pi z/P)}{\sin(\pi z/P)})^2.$$

This is included here not because we need it as much as because this expression is much easier to graph:

You see how these functions seem to be, as $M \to \infty$, approaching the spiky-looking Dirac delta function. In fact, (as distributions) they do. (A *distribution* is a linear functional on the vector space of all compactly supported infinitely differentiable functions $C_c^\infty(\mathbb{R})$.) In other words,

$$\lim_{M\to\infty}\int_{\mathbb{R}} f(x)K_M(x-t)\,dx = \int_{\mathbb{R}} f(x)\delta(x-t)\,dx = f(t).$$

This is the essential reason why, if f is continuous at x, $\lim_{M\to\infty} S_M^C(x) = f(x)$.

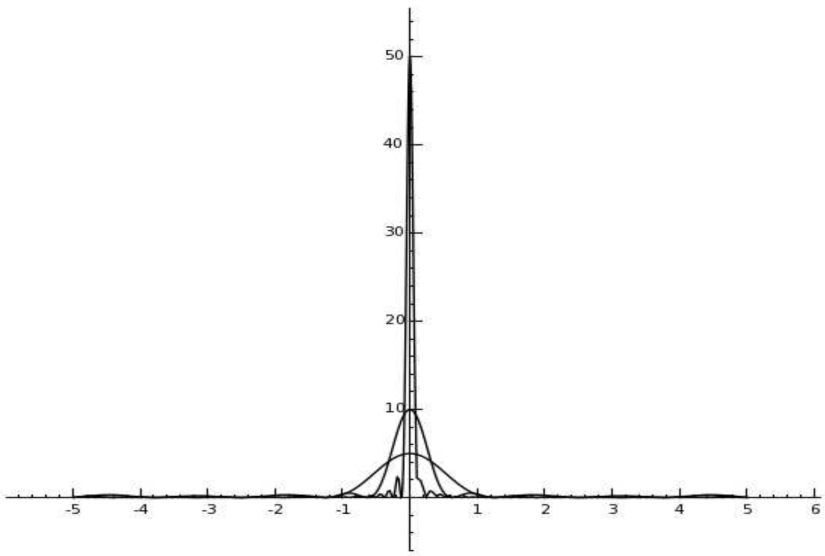


Figure 26: List plot of values of $2\pi K_M(x)$, $M = 5, 10, 50$, $P = 2\pi$ (normalized by 2π for simplicity).

The SAGE command for this:

```
sage: M = 5
sage: f = lambda z: (1/(M+1))*(sin((M+1)*z/2)/sin(z/2))^2
sage: P1 = plot(f,-5,5)
sage: M = 10
sage: f = lambda z: (1/(M+1))*(sin((M+1)*z/2)/sin(z/2))^2
sage: P2 = plot(f,-5,5)
sage: M = 50
sage: f = lambda z: (1/(M+1))*(sin((M+1)*z/2)/sin(z/2))^2
sage: P3 = plot(f,-5,5)
sage: show(P1+P2+P3)
```

4.3 The discrete analog

There is a discrete analog of the $C_M = C_{M,f}$ which may be regarded as a Fourier multiplier operator. This subsection is included for the purpose of illustrating the notion of a Fourier multiplier operator by means of an example (as you will see, our process of discretizing the filter ruins its usefulness).

If we replace the usual FT on $(0, P)$

$$f \longmapsto c_k = c_k(f) = \frac{1}{P}\int_0^P f(t)e^{-2\pi ikt/P}\,dt,$$

by the DFT on $\mathbb{C}^N$,

$$\vec{f} \longmapsto DFT(\vec{f})_k$$

then the Cesàro filter becomes simply the modification

$$\vec{f} \longmapsto (1-\frac{k}{N})DFT(\vec{f})_k.$$

If we define $\xi \in V_N$ by $\xi(k) = (1-\frac{k}{N})$ then, in functional notation, the Cesàro filter becomes the modification

$$f \longmapsto \xi f^{\wedge}, \qquad (f \in V_N).$$

We define the *Cesàro filter map* $C : V_n \to V_N$ by

$$C(f) = (\xi f^{\wedge})^{\vee}.$$

This operator C is, by construction, a Fourier multiplier operator: $C = M_\xi$. Here is an example computation.

Example 43 *Let $f(x) = 10x(1-x)$, $0 < x < 1$, which we sample at $N = 25$ regularly spaced points. The plot of this function is an inverted parabola, passing through 0 and 1 on the x-axis. Below, we plot both f and the real part of $C(f)$:*

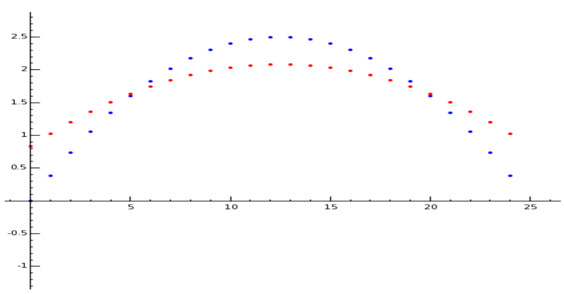


Figure 27: List plot of values of $10x(1-x)$ and (the real part of) its image under a FMO.

4.4 Hann filter

As usual, let $S_M(x) = \sum_{j=-M}^{M} c_k e^{2\pi ijx/P}$ denote the M-th partial sum of the FS of f and let

$$S_M^H(x) = \sum_{j=-M}^{M} H_M(j) c_j e^{2\pi ijx/P},$$

where $H_M(x) = (1+\cos(\frac{\pi x}{M}))/2$ is the *Hann filter*[8].

Definition 44 The *M-th Hann-filtered* partial sum of the FS of f is the function $S_M^H = S_{M,f}^H$ above.

Here is an integral representation for S_M^H. Since

$$c_j = \frac{1}{P}\int_0^P f(t)e^{-2\pi ijt/P}\,dt,$$

we have

$$\begin{array}{ll} S_M^H(x) & = \sum_{j=-M}^{M} H_M(j)\frac{1}{P}(\int_0^P f(t)e^{-2\pi ijt/P}\,dt)e^{2\pi ijx/P} \\ & = \frac{1}{P}\int_0^P f(t)\sum_{j=-M}^{M} H_M(j)e^{2\pi ij(x-t)/P}\,dt \\ & = \int_0^P f(t)K_M^H(x-t)\,dt, \end{array} \tag{22}$$

where

$$K_M^H(z) = \frac{1}{P}\sum_{j=-M}^{M} H_M(j)e^{2\pi ijz/P}.$$

This function is called the *Hann kernel.* This has a simpler expression,

$$K_M^H(z) = \frac{1}{4}(2K_M^D(z) + K_M^D(z+\frac{\pi}{M}) + K_M^D(z-\frac{\pi}{M})),$$

where K_M^D is the Dirichlet kernel.

[8]With $0.54 + 0.46\cos(\frac{\pi x}{M})$ instead of H_M, you get the *Hamming filter.* We shall not describe this due to its similarity with the Hann filter.

Example 45 Consider the odd function of period 2π defined by

$$f(x) = \begin{cases} -1, & 0 \leq x < \pi/2, \\ 2, & \pi/2 \leq x < \pi. \end{cases}$$

We use `SAGE` to compare the ordinary partial sum $S_{20}(x)$ to the Hann-filtered partial sum $S_{20}^H(x)$ and the Césaro-filtered partial sum S_{20}^C. As you can see from the graphs, the filtered partial sum is "smoother" than $S_{20}(x)$ and a slightly better fit than S_{20}^C.

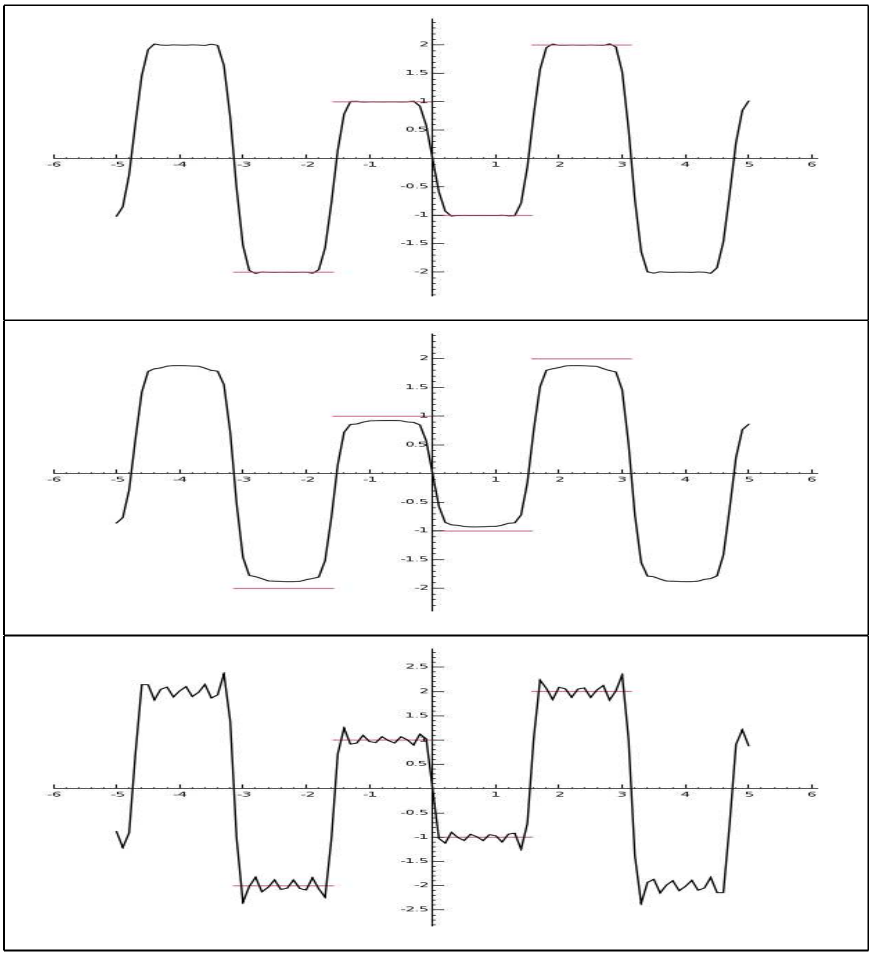


Figure 28: Plot of $f(x)$, $S_{20}^H(x)$. Plot of $f(x)$, $S_{20}^C(x)$. Plot of $f(x)$, $S_{20}(x)$.

The SAGE command for this:

```
sage: f1 = lambda x:-2
sage: f2 = lambda x:1
sage: f3 = lambda x:-1
sage: f4 = lambda x:2
sage: f = Piecewise([[(-pi,-pi/2),f1],[(-pi/2,0),f2],[(0,pi/2),f3],[(pi/2,pi),f4]])
sage: P1 = f.plot_fourier_series_partial_sum_hann(20,pi,-5,5)
sage: P2 = f.plot(rgbcolor=(0.8,0.3,0.4))
sage: P3 = f.plot_fourier_series_partial_sum(20,pi,-5,5)
sage: P4 = f.plot_fourier_series_partial_sum_cesaro(20,pi,-5,5)
sage: show(P1+P2+P3+P4)
```

4.5 Poisson summation formula

Let $\delta(x)$ be the Dirac delta function. The Poisson summation formula can be regarded as formula for the FT of the "Dirac comb":

$$\Delta(x) \;=\; \sum_{n=-\infty}^{\infty} \delta(x - Pn).$$

We shall not need this formulation, or any of the many very interesting generalizations of this formula. In the next section, it will be applied to proving the Shannon sampling theory, so we only present here what we need for that.

Theorem 46 *(Poisson summation) Assume that $P > 0$ is fixed and*

- *f be a continuous function,*
- *the function*

  $$f_P(x) = \sum_{n\in\mathbb{Z}} f(x - \frac{n}{P})$$

  converges uniformly on $[-P/2, P/2]$, and
- $$\sum_{n\in\mathbb{Z}} |\hat{f}(\frac{n}{P})|$$

  converges.

Then

$$\sum_{n\in\mathbb{Z}} f(x-\frac{n}{P}) = \frac{1}{P}\sum_{n\in\mathbb{Z}} \hat{f}(\frac{n}{P})e^{2\pi ix/P}.$$

proof: This theorem says that the $n-th$ Fourier series coefficient of the perioic function f_P is

$$c_n = \frac{1}{P}\hat{f}(\frac{n}{P}),$$

where here $\hat{f}$ denotes the Fourier transform (on $\mathbb{R}$). This identity is verified by the following computation:

$$\begin{array}{ll} c_n & = \frac{1}{P}\int_0^P f_P(x)e^{-2\pi ix/P}\,dx \\ & = \sum_{n\in\mathbb{Z}} \frac{1}{P}\int_0^P f(x-nP)e^{-2\pi ix/P}\,dx \\ & = \sum_{n\in\mathbb{Z}} \frac{1}{P}\int_{-nP}^{(n+1)P} f(x)e^{-2\pi ix/P}\,dx \\ & = \frac{1}{P}\int_{\mathbb{R}} f(x)e^{-2\pi ix/P}\,dx \\ & \frac{1}{P}\hat{f}(\frac{n}{P}), \end{array}$$

as desired. □

4.6 Shannon's sampling theorem

Let f be a continuous function periodic with period P. So far, we have started with sampling values of $f(x)$ at N equally spaces points to get a vector $\vec{f} = (f(\frac{j}{N}P))_{j=0,1,\dots,N-1}$, then computed its DFT, which we showed was a good approximation to the FS coefficients:

$$\begin{array}{ccc} f(x) & \longmapsto & \vec{f} \\ c_k = \frac{1}{P}\int_0^P f(x)e^{-2\pi ikx/P}\,dx & \approx & DFT(\vec{f})_k \\ f(x) = \sum_{k\in\mathbb{Z}} c_k e^{2\pi ikx/P} & \leftrightarrow & (\frac{1}{N}\sum_{j=o}^{N-1} DFT(\vec{f})_k W^{kj})_{k=0,1,\dots,N-1} \end{array}$$

Based on this, we might expect that $f(x)$ can be approximately reconstructed from its sample values alone. For example, perhaps something like $f(x) \overset{?}{\approx} \sum_{|k|<N} DFT(\vec{f})_k e^{2\pi ikx/P}$ might be true. Shannon's Sampling Theorem says that, under certain condtions, $f(x) \in L^1(\mathbb{R})$ is *completely determined* from its sampled values.

We say $f \in L^1(\mathbb{R})$ is *band limited* if there is a number $L > 0$ such that $\hat{f}(t) = 0$ for all t with $|t| > L$. When such an L exists and is choosen as small as possible, the number $2L$ is called the *Nyquist rate* and the number $\frac{1}{2L}$ is the *sampling period.*

Define the "sink" function

$$\mathrm{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

Theorem 47 *(Shannon's Sampling Theorem) Assume f is as in the above theorem and that $f \in L^1(\mathbb{R})$ is band limited with Nyquist rate $P = 2L$. Then*

$$f(x) = \sum_{n\in\mathbb{Z}} f(\frac{n}{2L})\mathrm{sinc}(2Lx - n).$$

proof: Let $\hat{f}_P$ be defined analogously to f_P above. The Poisson formula gives

$$\begin{aligned} \hat{f}_P(y) &= \textstyle\sum_{n\in\mathbb{Z}} \hat{f}(y - \frac{n}{P}) \\ &= \frac{1}{P} \textstyle\sum_{n\in\mathbb{Z}} \hat{\hat{f}}(\frac{n}{P})e^{2\pi i y/P} \\ &= \frac{1}{P} \textstyle\sum_{n\in\mathbb{Z}} f(-\frac{n}{P})e^{2\pi i y/P}, \end{aligned}$$

since $\hat{\hat{f}}(x) = f(-x)$. By hypothesis, $\hat{f}(y) = 0$ for $|y| > P/2$, so $\hat{f}_P(y)$ is merely a periodic extension of $\hat{f}$. Let

$$\chi_P(y) = \begin{cases} 1, & |y| \leq P/2, \\ 0, & |y| > P/2. \end{cases}$$

so, by hypothesis, $\hat{f}(y) = \chi_P(y)\hat{f}_P(y)$. Multiply both sides of

$$\hat{f}(y) = \chi_P(y)\hat{f}_P(y) = \frac{1}{P}\sum_{n\in\mathbb{Z}} f(-\frac{n}{P})\chi_P(y)e^{2\pi i y/P}$$

by $e^{2\pi i x y}$ and integrate over $-P/2 < y < P/2$. On one hand,

$$\int_{-P/2}^{P/2} \hat{f}(y)e^{2\pi i x y}\, dy = \int_{\mathbb{R}} \hat{f}(y)e^{2\pi i x y}\, dy = f(x),$$

and on the other hand,

$$\begin{array}{c}\int_{-P/2}^{P/2} \frac{1}{P} \sum_{n\in\mathbb{Z}} f(-\frac{n}{P})\chi_P(y) e^{2\pi i y/P} e^{2\pi i x y}\, dy \\ = \frac{1}{P}\sum_{n\in\mathbb{Z}} f(-\frac{n}{P}) \int_{-P/2}^{P/2} \chi_P(y) e^{2\pi i y/P} e^{2\pi i x y}\, dy \\ = \sum_{n\in\mathbb{Z}} f(-\frac{n}{P}) S(x+\frac{n}{P}),\end{array}$$

where

$$S(z) = \frac{1}{P}\int_{-P/2}^{P/2} e^{2\pi i z y}\, dy = \mathrm{sinc}(\pi z).$$

□

4.7 Aliasing

Obviously most functions are not band limited. When a function is not band limited but the right-hand side of the above "reconstruction formula" is used anyway, the error creates an effect called "aliasing." This also occurs when one uses the reconstruction formula for a smaple rate lower than the Nyquist rate.

Aliasing is a major concern in the analog-to-digital conversion of video and audio signals.

Theorem 48 *(Aliasing Theorem) Assume f is as in the above theorem and that the FT of $f \in L^1(\mathbb{R})$ satisfies*

$$|\hat{f}(y)| \leq \frac{A}{(1+|y|)^\alpha},$$

for some constants $A > 0$ and $\alpha > 0$. Then

$$f(x) = \sum_{n\in\mathbb{Z}} f(\frac{n}{2L})\mathrm{sinc}(2Lx - n) + E(x),$$

where $E(x)$ is an "error" bounded by

$$|E(x)| \leq \frac{4A}{\alpha\cdot(1+P/2)^\alpha}.$$

5 Discrete sine and cosine transforms

Recall that, given a differentiable, real-valued, periodic function $f(x)$ of period $P = 2L$, there are a_n with $n \geq 0$ and b_n with $n \geq 1$ such that $f(x)$ has (real) Fourier series

$$FS_{\mathbb{R}}(f)(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}[a_n \cos(\frac{2\pi nx}{P}) + b_n \sin(\frac{2\pi nx}{P})].$$

where

$$a_n = \frac{2}{P}\int_0^P f(x)\cos(\frac{2\pi nx}{P})\,dx = \frac{1}{L}\int_{-L}^{L} f(x)\cos(\frac{\pi nx}{L})\,dx,$$

and

$$b_n = \frac{2}{P}\int_0^P f(x)\sin(\frac{2\pi nx}{P})\,dx = \frac{1}{L}\int_{-L}^{L} f(x)\sin(\frac{\pi nx}{L})\,dx.$$

When f is even then the $b_n = 0$ and we call the FS a *cosine series*. When f is odd then the $a_n = 0$ and we call the FS a *sine series*. In either of these cases, it suffices to define f on the interval $0 < x < L$ instead of $0 < x < P$.

Let us first assume f is even and"discretize" the integral for the k-th coefficient of the cosine series and use that as a basis for defining the discrete cosine transform or DCT. Using the "left-hand Riemann sum" approximation for the integral using N subdivisions, we have

$$\begin{aligned} a_k &= \tfrac{2}{L}\int_0^L f(x)\cos(\tfrac{\pi kx}{L})\,dx \\ &\approx \tfrac{2}{L}\sum_{j=0}^{N-1} f(Lj/N)\cos(\tfrac{\pi kLj/N}{L})\left(\tfrac{L}{N}\right) \\ &= \tfrac{2}{N}\sum_{j=0}^{N-1} f(Lj/N)\cos(\pi kj/N). \end{aligned} \tag{23}$$

This motivates the following definition.

Definition 49 The *N-point discrete cosine transform* (or *DCT*) of the vector $\vec{f} = (f_0, ..., f_{N-1}) \in \mathbb{R}^N$ is

$$DCT(\vec{f})_k = \sum_{j=0}^{N-1} f_j \cos(\pi kj/N),$$

where $0 \leq k < N$.

This transform is represented by the $N \times N$ real symmetric matrix $(\cos(\pi kj/N))_{0\leq j,k\leq N-1}$.

Example 50 *When $N = 5$, we have*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \cos(\frac{\pi}{5}) & \cos(\frac{2\pi}{5}) & \cos(\frac{3\pi}{5}) & \cos(\frac{4\pi}{5}) \\ 1 & \cos(\frac{2\pi}{5}) & \cos(4\frac{4\pi}{5}) & \cos(\frac{6\pi}{5}) & \cos(\frac{8\pi}{5}) \\ 1 & \cos(\frac{3\pi}{5}) & \cos(\frac{6\pi}{5}) & \cos(\frac{9\pi}{5}) & \cos(\frac{12\pi}{5}) \\ 1 & \cos(\frac{4\pi}{5}) & \cos(\frac{8\pi}{5}) & \cos(\frac{12\pi}{5}) & \cos(\frac{16\pi}{5}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0.8090 & 0.3090 & -0.3090 & -0.8090 \\ 1 & 0.3090 & -0.8090 & -0.8090 & 0.3090 \\ 1 & -0.3090 & -0.8090 & 0.8089 & 0.3090 \\ 1 & -0.8090 & 0.3090 & 0.3090 & -0.8089 \end{pmatrix}$$

and

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \sin(\frac{\pi}{5}) & \sin(\frac{2\pi}{5}) & \sin(\frac{3\pi}{5}) & \sin(\frac{4\pi}{5}) \\ 0 & \sin(\frac{2\pi}{5}) & \sin(4\frac{4\pi}{5}) & \sin(\frac{6\pi}{5}) & \sin(\frac{8\pi}{5}) \\ 0 & \sin(\frac{3\pi}{5}) & \sin(\frac{6\pi}{5}) & \sin(\frac{9\pi}{5}) & \sin(\frac{12\pi}{5}) \\ 0 & \sin(\frac{4\pi}{5}) & \sin(\frac{8\pi}{5}) & \sin(\frac{12\pi}{5}) & \sin(\frac{16\pi}{5}) \end{pmatrix} = \begin{pmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.5877 & 0.9510 & 0.9510 & 0.5877 \\ 0.0000 & 0.9510 & 0.5877 & -0.5877 & -0.9510 \\ 0.0000 & 0.9510 & -0.5877 & -0.5878 & 0.9510 \\ 0.0000 & 0.5877 & -0.9510 & 0.9510 & -0.5878 \end{pmatrix}.$$

Here is the `SAGE` *code for producing the DCT above::*

```
sage: RRR = RealField(15)
sage: MS = MatrixSpace(RRR,5,5)
sage: r = lambda j: [RRR(cos(pi*j*k/5)) for k in range(5)]
sage: dct = MS([r(j) for j in range(5)])
```

Next, assume f is odd and"discretize" the integral for the k-th coefficient of the sine series and use that as a basis for defining the discrete sine transform or DST. Using the "left-hand Riemann sum" approximation for the integral using N subdivisions, we have

$$\begin{aligned} b_k &= \frac{2}{L}\int_0^L f(x)\sin(\frac{\pi kx}{L})\,dx \\ &\approx \frac{2}{L}\sum_{j=0}^{N-1} f(Lj/N)\sin(\frac{\pi kLj/N}{L})\big(\frac{L}{N}\big) \\ &= \frac{2}{N}\sum_{j=0}^{N-1} f(Lj/N)\sin(\pi kj/N). \end{aligned} \tag{24}$$

This motivates the following definition.

Definition 51 The *N-point discrete sine transform* (or *DST*) of the vector $\vec{f} = (f_0 = 0, f_1, ..., f_{N-1}) \in \mathbb{R}^N$ is

$$DST(\vec{f})_k = \sum_{j=1}^{N-1} f_j \sin(\pi kj/N),$$

where $0 \leq k < N$.

This transform is represented by the $N \times N$ real symmetric matrix $(\sin(\pi ikj/N))_{0\leq j,k\leq N-1}$. Since the 0-th coordinate is always $= 0$, since $\sin(0) = 0$, sometimes this is replaced by a map $DST : \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$, represented by the $(N-1) \times (N-1)$ real symmetric matrix $(\sin(\pi kj/N))_{1\leq j,k\leq N-1}$.

One difference between these definitions and the definition of the DFT is that here the N samples are taken from $(0, L)$, whereas for the DFT the N samples are taken from $(0, P)$.

Is there some way to compute the DCT and the DST in terms of the DFT of a function? Let $\vec{f} = (f_0, f_1, ..., f_N, ..., f_{2N-1})$ and compute, for $0 \leq k < N$,

$$\begin{aligned}
DFT(\vec{f})_k &= \sum_{j=0}^{2N-1} f_j e^{\pi ikj/N} \\
&= \sum_{j=0}^{N-1} f_j e^{\pi ikj/N} + (-1)^k \sum_{j=0}^{N-1} f_{N+j} e^{\pi ikj/N} \\
&= \sum_{j=0}^{N-1} (f_j + (-1)^k f_{N+j}) e^{\pi ikj/N} \\
&= \sum_{j=0}^{N-1} (f_j + (-1)^k f_{N+j}) \cos(\pi kj/N) \\
&\qquad + i \sum_{j=0}^{N-1} (f_j + (-1)^k f_{N+j}) \sin(\pi kj/N) \\
&= DCT(\vec{g} + (-1)^k \vec{h})_k + iDST(\vec{g} + (-1)^k \vec{h})_k,
\end{aligned}$$

where $\vec{g} = (f_0, f_1, ..., f_{N-1})$ and $\vec{h} = (f_N, f_{N+1}, ..., f_{2N-1})$. Here the DFT is based on $2N$ sample values, whereas the DCT and DST are each based on N sample values.

Let $\vec{g} = (g_0, g_1, ..., g_{N-1}) \in \mathbb{R}^N$ and let $\vec{g}_* = (g_0, g_1, ..., g_{N-1}, 0, ..., 0) \in \mathbb{R}^{2N}$ denote its "extension by zero" to $\mathbb{R}^{2N}$. The above computation implies

$$DCT(\vec{g})_k = \mathrm{Re}(DFT(\vec{g}_*)_k), \qquad DST(\vec{g})_k = \mathrm{Im}(DFT(\vec{g}_*)_k),$$

for $0 \leq k < N$. In particular, if we can find a "fast" way of computing the DFT then we can also compute the DCT and the DST quickly. We turn to such efficient computational procedures inthe next section.

6 Fast Fourier transform

The fast Fourier transform is a procedure for computing the discrete Fourier transform which is especially fast. The term FFT often loosely refers to a hybrid combination of the two algorithms presented in this section.

The algorithm described first, due to James Cooley and John Tukey [CT], works when the number of sample values N is a power of 2, say $N = 2^r$, for some integer $r > 1$. This special case is also referred to as the *radix-2 algorithm*. This is the one we will describe in the next section.

6.1 Cooley-Tukey algorithm (radix-2)

First, we assume that the powers of $\overline{W}$ (namely, 1, $\overline{W}$, $\overline{W}^2$, ..., $\overline{W}^{n-1}$) have been precomputed. Note that the computation of the DFT on $\mathbb{C}^N$ requires N^2 multiplications. This is because the matrix F_N is $N \times N$ and each matrix entry is involved in the computation of the vector $DFT(\vec{f}) \in \mathbb{C}^N$. If $M(N)$ denotes the number of multiplications required to compute the DFT then the above reasoning shows that

$$M(N) \leq N^2.$$

To improve on this, the *Cooley-Tukey procedure* is described next. Let $N = 2M$ for the argument below. To be clear about the notation, let $W_N = e^{2\pi i/N}$, so $\overline{W}_N^2 = \overline{W}_M$. Let $\vec{f} = (f_0, f_1, ..., f_{N-1}) \in \mathbb{C}^N$ be the vector we want to compute the DFT of and write

$$\vec{f}_{even} = (f_0, f_2, ..., f_{N-2}) \in \mathbb{C}^M, \quad \vec{f}_{odd} = (f_1, f_3, ..., f_{N-1}) \in \mathbb{C}^M.$$

We have, for $0 \leq k < N$,

$$\begin{array}{ll} DFT(\vec{f})_k & = \sum_{j=0}^{N-1} f_j \overline{W}_N^{jk} \\ & = \sum_{j=0}^{M-1} f_{2j} \overline{W}_N^{2jk} + \sum_{j=0}^{M-1} f_{2j+1} \overline{W}_N^{(2j+1)k} \\ & = \sum_{j=0}^{M-1} f_{2j} \overline{W}_M^{jk} + \overline{W}_N^k \sum_{j=0}^{M-1} f_{2j+1} \overline{W}_M^{jk} \\ & = DFT(\vec{f}_{even})_k + \overline{W}_N^k DFT(\vec{f}_{odd})_k. \end{array} \tag{25}$$

Theorem 52 *For each $N = 2^L$, $M(N) \leq N \cdot (L+2)$.*

proof: We prove this by mathematical induction on L. This requires proving a (base case) step $N = 4$ and a step where we assume the truth of the inequality for $N/2$ and prove it for N.

The number of multiplications required to compute the DFT when $N = 4$ is ≤ 16. Therefore, $M(4) \leq 4 \cdot (2+2)$.

Now assume $M(N/2) \leq \frac{N}{2} \cdot (L+1)$. The Cooley-Tukey procedure (25) shows that $M(N) \leq 2 \cdot M(N/2) + N/2$. This and the induction hypothesis implies

$$M(N) \leq 2 \cdot M(N/2) + N/2 \leq 2(\frac{N}{2} \cdot (L+1)) + N/2 = (L+1+1/2)N.$$

This is $\leq N \cdot (L+2)$. $\square$

Example 53 *When $N = 8$, the DFT matrix is given by*

$$F_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta_8 & \zeta_8^2 & \zeta_8^3 & -1 & -\zeta_8 & -\zeta_8^2 & -\zeta_8^3 \\ 1 & \zeta_8^2 & -1 & -\zeta_8^2 & 1 & \zeta_8^2 & -1 & -\zeta_8^2 \\ 1 & \zeta_8^3 & -\zeta_8^2 & \zeta_8 & -1 & -\zeta_8^3 & \zeta_8^2 & -\zeta_8 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\zeta_8 & \zeta_8^2 & -\zeta_8^3 & -1 & \zeta_8 & -\zeta_8^2 & \zeta_8^3 \\ 1 & -\zeta_8^2 & -1 & \zeta_8^2 & 1 & -\zeta_8^2 & -1 & \zeta_8^2 \\ 1 & -\zeta_8^3 & -\zeta_8^2 & -\zeta_8 & -1 & \zeta_8^3 & \zeta_8^2 & \zeta_8 \end{pmatrix}$$

and the DFT matrix for $N/2 = 4$ is given by

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \zeta_4 & -1 & -\zeta_4 \\ 1 & -1 & 1 & -1 \\ 1 & -\zeta_4 & -1 & \zeta_4 \end{pmatrix}.$$

Let $\vec{f} = (0,1,2,3,4,5,6,7)$, so $\vec{f}_{even} = (0,2,4,6)$, $\vec{f}_{odd} = (1,3,5,7)$. We compute

$$F_8\vec{f} = \begin{pmatrix} 28 \\ -4\zeta_8^3 - 4\zeta_8^2 - 4\zeta_8 - 4 \\ -4\zeta_8^2 - 4 \\ -4\zeta_8^3 + 4\zeta_8^2 - 4\zeta_8 - 4 \\ -4 \\ 4\zeta_8^3 - 4\zeta_8^2 + 4\zeta_8 - 4 \\ 4\zeta_8^2 - 4 \\ 4\zeta_8^3 + 4\zeta_8^2 + 4\zeta_8 - 4 \end{pmatrix}.$$

Let Z denote the diagonal matrix with ζ_8^k's on the diagonal, $0 \leq k \leq 7$. (This matrix represents the factors $\overline{W}_N^k$ in the formula (25) above.) It turns out to be more useful for computations to split this matrix up into two parts:

$$Z_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \zeta_8 & 0 & 0 \\ 0 & 0 & \zeta_8^2 & 0 \\ 0 & 0 & 0 & \zeta_8^3 \end{pmatrix}, \qquad Z_2 = \begin{pmatrix} \zeta_8^4 & 0 & 0 & 0 \\ 0 & \zeta_8^5 & 0 & 0 \\ 0 & 0 & \zeta_8^6 & 0 \\ 0 & 0 & 0 & \zeta_8^7 \end{pmatrix}.$$

Note $Z_2 = -Z_1$. For the first 4 coordinates, we use

$$F_4\vec{f}_{even} + Z_1 F_4 \vec{f}_{odd} = \begin{pmatrix} 28 \\ -4\zeta_8^3 - 4\zeta_8^2 - 4\zeta_8 - 4 \\ -4\zeta_8^2 - 4 \\ -4\zeta_8^3 + 4\zeta_8^2 - 4\zeta_8 - 4 \end{pmatrix}$$

and for the last 4 coordinates, we use

$$F_4\vec{f}_{even} + Z_2 F_4 \vec{f}_{odd} = \begin{pmatrix} -4 \\ 4\zeta_8^3 - 4\zeta_8^2 + 4\zeta_8 - 4 \\ 4\zeta_8^2 - 4 \\ 4\zeta_8^3 + 4\zeta_8^2 + 4\zeta_8 - 4 \end{pmatrix}$$

We see that, as (25) predicts, $F_8\vec{f}$ is equal to $[F_4\vec{f}_{even} + Z_1F_4\vec{f}_{odd}, F_4\vec{f}_{even} + Z_2F_4\vec{f}_{odd}]$.

Here is the `SAGE` *code for the above computations.*

```
sage: MS4 = MatrixSpace(CyclotomicField(4),4,4)
sage: MS8 = MatrixSpace(CyclotomicField(8),8,8)
sage: V4 = VectorSpace(CyclotomicField(4),4)
sage: V8 = VectorSpace(CyclotomicField(8),8)
sage: z4 = CyclotomicField(4).gen()
sage: z8 = CyclotomicField(8).gen()
sage: r4 = lambda k: [z4^(j*k) for j in range(4)]
sage: r8 = lambda k: [z8^(j*k) for j in range(8)]
sage: F4 = MS4([r4(k) for k in range(4)])
sage: F8 = MS8([r8(k) for k in range(8)])
sage: f = V8([0,1,2,3,4,5,6,7])
sage: fe = V4([0,2,4,6])
sage: fo = V4([1,3,5,7])
sage: FFTe = [(F4*fe)[j]+z8^j*(F4*fo)[j] for j in range(4)]
sage: FFTo = [(F4*fe)[j]-z8^j*(F4*fo)[j] for j in range(4)]
sage: FFTe+FFTo

[28,
 -4*zeta8^3 - 4*zeta8^2 - 4*zeta8 - 4,
 -4*zeta8^2 - 4,
```

```
 -4*zeta8^3 + 4*zeta8^2 - 4*zeta8 - 4,
 -4,
 4*zeta8^3 - 4*zeta8^2 + 4*zeta8 - 4,
 4*zeta8^2 - 4,
 4*zeta8^3 + 4*zeta8^2 + 4*zeta8 - 4]
sage: [(F8*f)[j] for j in range(8)]

[28,
 -4*zeta8^3 - 4*zeta8^2 - 4*zeta8 - 4,
 -4*zeta8^2 - 4,
 -4*zeta8^3 + 4*zeta8^2 - 4*zeta8 - 4,
 -4,
 4*zeta8^3 - 4*zeta8^2 + 4*zeta8 - 4,
 4*zeta8^2 - 4,
 4*zeta8^3 + 4*zeta8^2 + 4*zeta8 - 4]
```

Finally, we give an example which only illustrates SAGE 's implementation of the FFT (which calls functions in the GSL [GSL]), as compared to its implementation of its DFT (which is implemented in Python but calls Pari for the computations involving N-th roots of unity over $\mathbb{Q}$):

```
sage: J = range(30)
sage: A = [QQ(int(10*(random()-1/2))) for i in J]
sage: s = IndexedSequence(A,J)
sage: time dfts = s.dft()
CPU times: user 0.86 s, sys: 0.04 s, total: 0.90 s
Wall time: 0.94
sage: time ffts = s.fft()
CPU times: user 0.00 s, sys: 0.00 s, total: 0.00 s
Wall time: 0.00
sage: J = range(3000)
sage: A = [QQ(int(10*(random()-1/2))) for i in J]
sage: s = IndexedSequence(A,J)
sage: time ffts = s.fft()
CPU times: user 0.21 s, sys: 0.00 s, total: 0.21 s
Wall time: 0.21
```

As you can see, for a sample vector in $\mathbb{C}^{3000}$, SAGE can compute the FFT in about $\frac{1}{5}$-th of a second. However, of you try to compute the DFT using this example, SAGE will probably give you an error related to its extreme size.

6.2 Rader's algorithm

In this subsection, we assume N is prime. Here we breifly describe an algorithm due to Rader [Ra] for computing the DFT on V_N for prime N.

The basic idea is to rewrite the DFT on V_N as a convolution on V_{N-1}. Remark 1 is then used to show that this convolution can be computed using a "fast" algorithm.

The first step in the algorithm is to select an element g, $1 < g < N-1$, which has the property that every element y in $\{1, 2, ..., N-1\}$ can be written in the form $y = g^x \pmod N$ for some x. This element g s called a *primitive root* $\pmod N$ (or a *generator of* $(\mathbb{Z}/N\mathbb{Z})^\times$), where $(\mathbb{Z}/N\mathbb{Z})^\times = \mathbb{Z}/N\mathbb{Z} - \{0\}$. Here is a table of primitive roots for various small primes N, and a demonstration, in the case $N = 17$, that $g = 3$ is indeed a primitive root:

p	3	5	7	11	11	13	17	19	23
g	2	2	3	2	2	2	3	2	5

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$g^k \pmod N$	1	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6

The `SAGE` command which produces the smallest primitive root $\pmod N$ is `primitive_root(N)`. For example, the above tables were produced using the commands

```
sage: [primitive_root(p) for p in [3,5,7,11,13,17,19,23]]
 [2, 2, 3, 2, 2, 3, 2, 5]
sage: N = 17; g = 3
sage: [g^k%N for k in range(N-1)]
[1, 3, 9, 10, 13, 5, 15, 11, 16, 14, 8, 7, 4, 12, 2, 6]
```

The next step in the algorithm is to rewrite the DFT on V_N as a convolution on V_{N-1}. To this end, fix $f \in V_N$ and define $h_1, h_2 \in V_{N-1}$ as

$$h_1(x) = f(g^{-x}), \qquad h_2(x) = e^{-2\pi i g^x/N},$$

for $x \in \{0, 1, 2, ..., N-2\} = \mathbb{Z}/(N-1)\mathbb{Z}$. For $k = 0$ we compute

$$DFT(f)_k = DFT(f)_0 = f(0) + f(1) + ... + f(N-1).$$

For $\ell \neq 0$, we let $m = m(\ell)$ denote the element of $\{0, 1, 2, ..., N-2\}$ such that $\ell = g^{-m}$. For $k \neq 0$, we let $n = n(k)$ denote the element of $\{0, 1, 2, ..., N-2\}$ such that $k = g^n$. Now write

$$\begin{aligned} DFT(f)_k &= \sum_{\ell=0}^{N-1} f(\ell) e^{-2\pi i k\ell/N} \\ &= f(0) + \sum_{\ell=1}^{N-1} f(\ell) e^{-2\pi i k\ell/N} \\ &= f(0) + \sum_{m=0}^{N-2} f(g^{-m}) e^{-2\pi i g^{n-m}/N} \\ &= f(0) + \sum_{m=0}^{N-2} h_1(m) h_2(n-m) \\ &= f(0) + h_1 * h_2(n). \end{aligned}$$

This is a convolution on V_{N-1}. If $N-1$ is a power of 2 (e.g., for $N = 17$) then use Remark 1 and the radix-2 Cooley-Tukey algorithm described in the previous section to quickly compute $h_1 * h_2$. If $N-1$ is not a power of 2, the best thing to do is to let P denote the smallest power of 2 greater than $N-1$ and extend both h_1 and h_2 by 0 to the range $\{0, 1, 2, ..., P-1\}$ (this is called "padding" the functions). Call these new functions $\tilde{h}_1$ and $\tilde{h}_2$. Using $DFT(f)_k = f(0) + \tilde{h}_1 * \tilde{h}_2(n)$ (where $k = g^n$). We now have expressed the DFT on V_N in terms of a convolution on V_P, where $P \leq 2N$. By Remark 1, we know that this can be computed in $\leq cN \log_2(N)$ multiplications, for some constant $c \geq 1$ (in fact, $c = 4$ should work if N is sufficiently large).

7 Fourier optics

Fourier optics is a special topic in the theory of optics. Good references are Goodman [Go] and chapter 5 of Walker [W1]. This section owes much to discussions with Prof. Larry Tankersley of the USNA Physics dept.

The object of this section is to describe a method for computing certain quantities arising in defraction experiments. I'll try to describe these experiments next.

Consider a monochromatic light source having wavelength λ. For example, the light emitted from a laser would fit this[9]. Imagine x- and y-coordinates in the aperture plane. The aperture function $A(x, y)$ is defined

[9] In reality, a laser beam is too narrow, so a series of lenses is required to widen it and then straighten out the light rays. You need to make sure that the beam is wide enough for it to be possible to place a small aperture (a slit or defraction grating, for instance) in front of it to allow only the light through that the aperture.

to be 1 if light can pass though the "slit" at (x, y) and 0 otherwise[10]. The light which passes through the aperture is pictured on a screen. It is this pattern which we wish to describe in this section.

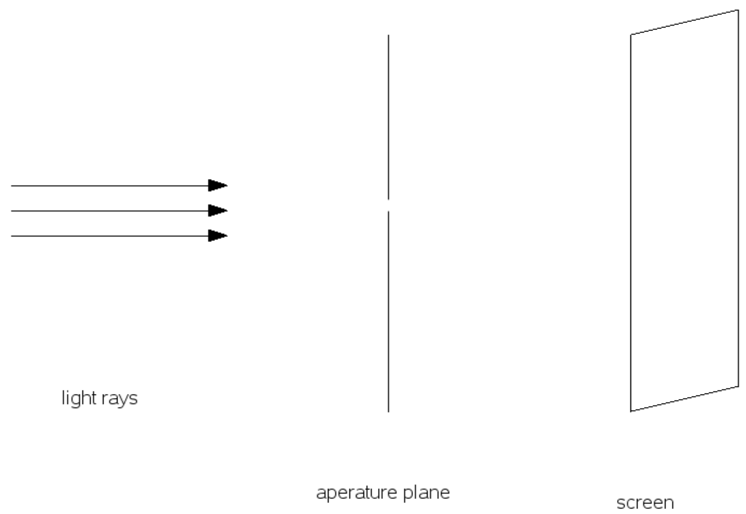


Figure 29: Slit experiment set-up for Fourier optics model.

[10](It is even possible to image values of $A(x, y)$ in the range between 0 and 1 representing a partially opaque screen, though we shall not need this here. However, we do assume A is real-valued.

When the aperture is a square (whose sides are aligned parallel to the x- and y-axes), the image projected on the screen of this experiment looks like a dotted plus sign, where the dots get fainter and fainter as they move away from the center. A square slit diffraction pattern is pictured below[11]:

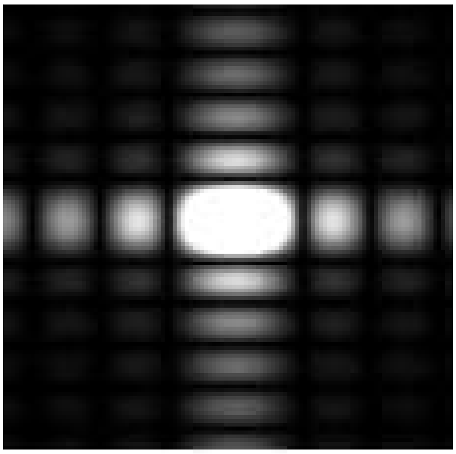


Figure 30: Square aperture diffraction experiment, Betzler [B].

The goal of this last section will be to describe the mathematics behind this "dotted plus sign" square slit diffraction pattern.

7.1 The mathematical model

The theory we shall sketch is called scalar defraction theory. A special case which we shall concentrate on is the Fraunhofer defraction model.

Let L denote the distance from the aperture to the screen, λ (as above) the wavelength of the light, and $a > 0$ the width of the slit. The *Fresnel number*[12] is the quantity $F = \frac{a^2}{\lambda L}$. When $F \geq 1$ the screen is "sufficiently far" (past the "Fresnel threshold") from the aperture that the wavefronts created by the slit have negligable curvature.

[11]This image is copyright Prof. Dr. Klaus Betzler and reproduced with his kind permission (stated in an email dated May 2, 2007)

[12]Pronounced "Fre-nell".

Here is Wikipedia's image[13] [WFd]:

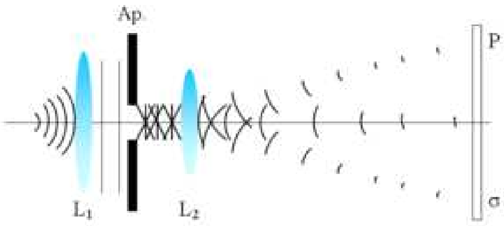


Figure 31: On this diagram, a wave is diffracted and observed at point σ. As this point is moved further back, beyond the Fresnel threshold, Fraunhofer diffraction occurs.

We also assume that the index of refraction of the medium is 1. If a light ray of wavelength λ travels from P to Q, points at a distance of $r = ||Q-P||$ from each other, at time t then the light amplitude $\psi(Q,t)$ at Q arising from this light ray satisfies the property

$$\psi(Q,t) = \psi(P,t)\frac{e^{\frac{2\pi i r}{\lambda}}}{\lambda r},$$

where $\psi(P,t)$ is the light amplitude at P. We can and do assume $P = (x,y,0)$ is a point in the aperture plane and $Q = (x',y',L)$ is a point on the screen. We assume $||Q-P||$ is not too big - so that light travels essentially instantaneously from P to Q. The superposition principle implies that the total light amplitude at Q satisfies

$$\psi(Q,t) = \int_{\mathbb{R}^2} A(P)\psi(P,t)\frac{e^{\frac{2\pi i||Q-P||}{\lambda}}}{\lambda||Q-P||}\,dP, \tag{26}$$

where we have identified the plane of the aperture slit with the Cartesian plane $\mathbb{R}^2$. If the distance between the aperture and he screen is large enough then r is a constant

The light *intensity* is defined by

[13]Wikipedia's images are copyright the Wikipedia Foundation and distributed under the GNU Documentation License.

$$I(Q) = \lim_{T\to\infty} \frac{1}{T}\int_0^T |\psi(Q,t)|^2\, dt.$$

The light amplitude is not typically measurable (at least not in air, with present day technology), but the intensity is.

Let P and P' be points on the aperture plane where $A(P) \neq 0$ and $A(P') \neq 0$. The *coherency function* is defined by

$$\Gamma(P,P') = \lim_{T\to\infty} \frac{1}{\sqrt{I(P)I(P')}}\frac{1}{T}\int_0^T \psi(P,t)\overline{\psi(P',t)}\, dt.$$

We say thet the light is *coherent* if $\Gamma(P,P') = 1$ for all such P, P'.

Using (26), we have

$$\begin{array}{ll} I(Q) & = \lim_{T\to\infty} \frac{1}{T}\int_0^T \psi(Q,t)\overline{\psi(Q,t)}\, dt \\ & = \lim_{T\to\infty} \frac{1}{T}\int_0^T (\int_{\mathbb{R}^2} A(P)\psi(P,t)\frac{e^{\frac{2\pi i||Q-P||}{\lambda}}}{\lambda||Q-P||}\, dP)\times \\ & \quad \times(\int_{\mathbb{R}^2} A(P')\overline{\psi(P',t)}\frac{e^{-\frac{2\pi i||Q-P'||}{\lambda}}}{\lambda||Q-P'||}\, dP')\, dt \\ & = \lim_{T\to\infty} \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \frac{A(P)A(P')}{\lambda^2||Q-P||||Q-P'||}e^{\frac{2\pi i(||Q-P||-||Q-P'||)}{\lambda}} \times \\ & \quad \times(\frac{1}{T}\int_0^T \psi(P,t)\overline{\psi(P',t)}dP\, dP')\, dt. \end{array}$$

Assuming that the light is coherent, this is

$$\begin{array}{l} = \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \frac{A(P)A(P')}{\lambda^2||Q-P||||Q-P'||}e^{\frac{2\pi i(||Q-P||-||Q-P'||)}{\lambda}} \sqrt{I(P)I(P')}\, dP\, dP' \\ = |\int_{\mathbb{R}^2} \frac{A(P)}{\lambda||Q-P||}e^{\frac{2\pi i||Q-P||}{\lambda}} \sqrt{I(P)}\, dP|^2. \end{array}$$

We assume[14] that $I(P) = 1$ for all points P on the aperture screen with $A(P) \neq 0$. In this case, the above reasoning leads to

$$I(Q) = |\int_{\mathbb{R}^2} \frac{A(P)}{\lambda||Q-P||}e^{\frac{2\pi i||Q-P||}{\lambda}}\, dP|^2. \qquad (27)$$

This is the key equation in scalar diffraction theory that enables one to approximate the intensity function in terms of the Fourier transform of the aperture function.

[14]The assumption that the intensity is constant at the aperture is all that is really necessary. We assume that this constant is $= 1$ for simplicity.

7.2 The Fraunhofer model

Notation: In x, y, z-coordinates, the aperature plane is described by $z = 0$, the screen is described by $z = L > 0$. In the diagram below, if $Q = (x, y, L)$ then $Q' = (x', y', L)$, $P' = (x, y, 0)$, and $P = (x', y', 0)$.

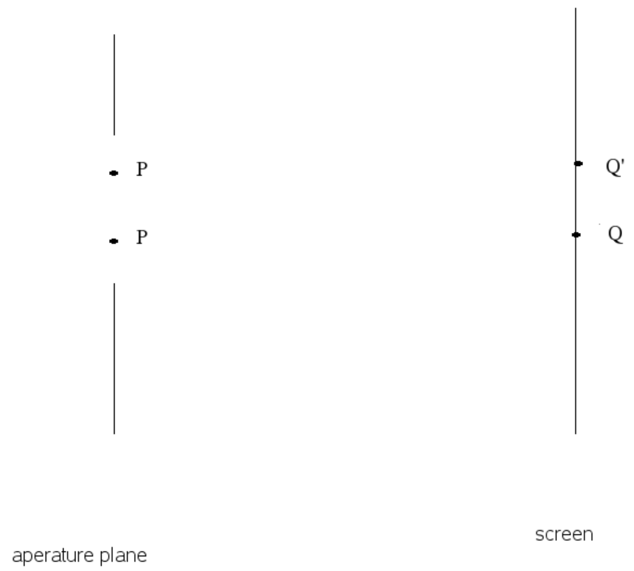


Figure 32: Notation for slit experiment set-up for Fraunhofer model.

In particular, the vector $\vec{v} = Q - P'$ is orthogonal to the vector $\vec{w} = P' - P$ and $P \cdot P' = P \cdot Q$. These give us

$$\begin{aligned} ||Q - P|| = ||Q - P' + P' - P|| &= (||Q - P'||^2 + ||P' - P||^2)^{1/2} \\ &= L \cdot (1 + \tfrac{||P'-P||^2}{L^2})^{1/2} = L + \tfrac{||P'-P||^2}{2L} + \ldots \ , \end{aligned}$$

by the power series expansion $(1 + x)^{1/2} = 1 + \frac{1}{2}x + \ldots.$ In addition to the coherence assumption, we also assume that L is so large and the aperture opening (i.e., the support of the aperture function A in the aperture plane, which we identify with $\mathbb{R}^2$) is so small that the error in

$$e^{\frac{2\pi i ||Q-P||}{\lambda}} \approx e^{\frac{\pi i ||P'-P||^2}{L\lambda}} e^{\frac{2\pi i L}{\lambda}}$$

is negligable. We expand $||P' - P||^2 = ||P'||^2 - 2P' \cdot P + ||P||^2$, so

$$e^{\frac{2\pi i||Q-P||}{\lambda}} \approx e^{\frac{\pi i||P||^2}{L\lambda}} e^{\frac{2\pi i L}{\lambda}} e^{\frac{\pi i||P'||^2}{L\lambda}} e^{\frac{2\pi i P\cdot P'}{L\lambda}} = e^{\frac{\pi i||P||^2}{L\lambda}} e^{\frac{2\pi i L}{\lambda}} e^{\frac{\pi i||P'||^2}{L\lambda}} e^{\frac{2\pi i P\cdot Q}{L\lambda}}.$$

If $f \in L^1(\mathbb{R}^2)$, define the *2-dimensional Fourier transforms* by

$$\hat{f}(u,v) = \int_{\mathbb{R}^2} f(x,y) e^{-2\pi i x u - 2\pi i y v}\, dxdy.$$

We also assume that L is so large and the aperture opening is so small that the error in $e^{\frac{\pi i||P||^2}{L\lambda}} \approx 1$ is negligable. Plugging this into (27), therefore gives

$$I(Q) \approx |\int_{\mathbb{R}^2} \frac{A(P)}{\lambda||Q-P||} e^{\frac{2\pi i P\cdot Q}{L\lambda}}\, dP|^2 \approx |\frac{1}{\lambda L}\int_{\mathbb{R}^2} A(P) e^{\frac{2\pi i P\cdot Q}{L\lambda}}\, dP|^2. \qquad (28)$$

In otherwords, if $Q = (x,y,L)$ then $I(Q) = \frac{1}{\lambda^2 L^2}|\hat{A}(\frac{x}{L\lambda}, \frac{y}{L\lambda})|^2$, where $\hat{A}$ denotes the 2-d Fourier transform.

Example 54 *We consider the example where the slit is a small rectangle. For example, suppose*

$$A(x,y) = \chi_{-\epsilon_1,\epsilon_1}(x)\chi_{-\epsilon_2,\epsilon_2}(y),$$

where $\epsilon_1 > 0$ *and* $\epsilon_2 > 0$ *are given, and* $\chi_{a,b}$ *represents the function which is* 1 *for* $a \leq x <\leq b$ *and* 0*, otherwise.*

We know

$$\int_{-a}^{a} e^{2\pi i x u}\, dx = \frac{\sin(2\pi a u}{\pi a u} 2\text{sinc}(2\pi a u),$$

therefore

$$\hat{A}(\frac{x}{L\lambda}, \frac{y}{L\lambda}) = 4\text{sinc}(2\pi(\frac{x}{L\lambda})\text{sinc}2\pi(\frac{y}{L\lambda}),$$

and so

$$I(Q) = \frac{4}{\lambda^2 L^2}(\text{sinc}(2\pi\epsilon_1 \frac{x}{L\lambda})\text{sinc}(2\pi\epsilon_2 \frac{y}{L\lambda}))^2.$$

Here are the SAGE commands which produce the plot below for this intensity function (the axes have been scaled for simplicity).

```
sage: f = "(sin(x)/x)^2*(sin(y)/y)^2"
sage: opts = "[plot_format, openmath]"
sage: maxima.plot3d (f, "[x, -5, 5]", "[y, -5, 5]", opts)
```

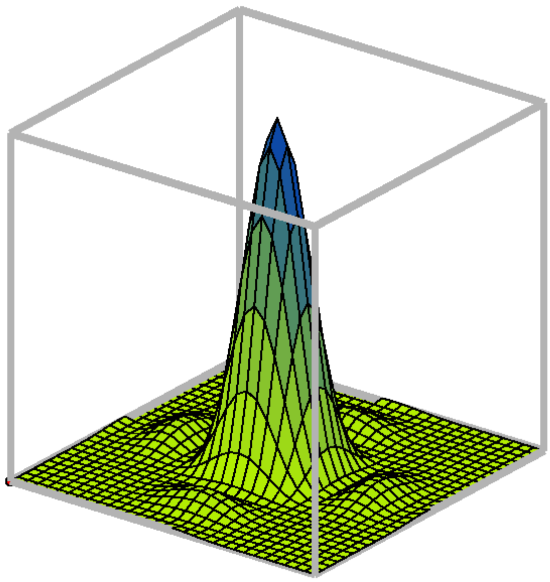


Figure 33: Intensity function for a square slit experiment in the Fraunhofer model.

This plot is consistent with the image pictured in the "dotted plus sign" square slit diffraction pattern. As the mathematical explanation of this image was the goal of this last section of these notes, we are done.

8 Additional reading

For an elementary introduction to wavelets, appropriate for capstone projects, see Frazier [F].

A number of papers from the American Math. Monthly in the theory of FS is available at the URL

http://math.fullerton.edu/mathews/c2003/FourierTransformBib/Links/FourierTransformBib_lnk_2.html

A number of those articles would also make good starters for a capstone project.

Finally, we recommend Walker [W2] and Körner [K] as excellent additional references.

References

[B] K. Betzler, *Fourier optics in examples*, Universität Osnabrück lecture notes,
http://www.home.uos.de/kbetzler/notes/

[CT] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. **19**(1965) 297-301.

[F] M. W. Frazier, *An introduction to wavelets through linear algebra*, Springer, 1999.

[GSL] Gnu Scientific Library of mathematical functions, written in C,
http://www.gnu.org/software/gsl/

[G] I. J. Good, "Analogues of Poisson's summation formula," Amer. Math. Monthly **69**(1962)259-266.

[Go] J. Goodman, *Introduction to Fourier optics*, McGraw-Hill, 1968.

[K] T. W. Körner, *Fourier analysis*, Cambridge Univ. Press, 1989.

[MP] J. McClellan, T. Parks, "Eigenvalue and eigenvector decomposition of the discrete Fourier transforms," IEEE Trans. Audio and Electroacustics **AU-20**(1972)66-72.

[Ra] C. Rader, "Discrete Fourier transforms when the number of data smaples is prime," Proc. IEEE **56**(1968)1107-1108.

[R] A. Robert, "Fourier series of polygons," The American Mathematical Monthly, Vol. 101, No. 5. (May, 1994), pp. 420-428.

[S] W. Stein, SAGE, free and open-source software available at http://sage.math.washington.edu/sage/.

[T] E. C. Titchmarsh, *Theory of functions*, Oxford University Press.

[W1] James Walker, *Fast Fourier transform*, 2nd ed, CRC Press, 1996.

[W2] ——, *Fourier Analysis* Oxford University Press, Oxford, 1988.

[WFd] *Fraunhofer diffraction*, Wikipedia
http://en.wikipedia.org/wiki/Fraunhofer_diffraction

Prof. W. D. Joyner, Math Dept, USNA, wdj@usna.edu
Last updated: 5-2-2007.