3. Fourier modes and Fourier series

`ATAPformats`

Throughout applied mathematics, one encounters three closely analogous canonical settings associated with the names of Fourier, Laurent, and Chebyshev. In fact, if we impose certain symmetries in the Fourier and Laurent cases, the analogies become equivalences. The Fourier setting is the one of central interest in this book, concerning a variable θ and a function f defined on $[-\pi, \pi]$:

$$\textit{Fourier:} \quad \theta \in [-\pi, \pi], \quad f(\theta) \approx \sum_{k=-n}^{n} a_k e^{ik\theta}. \tag{3.1}$$

Here $e^{ik\theta}$ is the kth Fourier mode, which we shall discuss in a moment. For the equivalent Laurent problem, let z be a variable that ranges over the unit circle in the complex plane. Given $f(\theta)$, define a transplanted function $F(z)$ on the unit circle by the condition $F(z) = f(\theta)$, where $z = e^{i\theta}$ as in (2.1) or $\theta = -i \log z$. Note that this means that there is a unique value of z for each value of θ. The series now involves a polynomial in both z and z^{-1}, known as a *Laurent polynomial:*

$$\textit{Laurent:} \quad |z| = 1, \quad F(z) \approx \sum_{k=-n}^{n} a_k z^k. \tag{3.2}$$

For the equivalent Chebyshev problem, let x be a variable that ranges over $[-1, 1]$ and assume that f is an even function of θ: $f(\theta) = f(-\theta)$, which imples $F(z) = F(z^{-1})$. Transplant f and F to a function $\mathcal{F}$ defined on $[-1, 1]$ by setting $\mathcal{F}(x) = F(z) = f(\cos\theta)$ as in (2.2). Now we have a 1-to-1 correspondence $z = e^{i\theta}$ between θ and z and a 2-to-1 correspondence between θ and x, with the symmetry $\mathcal{F}(\theta) = \mathcal{F}(-\theta)$, and the series is a Chebyshev polynomial:

$$\textit{Chebyshev:} \quad x \in [-1, 1], \quad \mathcal{F}(x) \approx \sum_{k=0}^{n} a_k \cos(k \arccos(x)). \tag{3.3}$$

One can carry (3.1)–(3.3) further by introducing canonical systems of grid points in the three settings. We have already seen the n equispaced points

$$\textit{Equispaced points:} \quad \theta_j = -\pi + 2\pi j/n, \quad 0 \le j \le n-1. \tag{3.6}$$

and we have interpreted these in terms of the (n)th roots of unity rotated by an angle $-\pi$ or π:

$$\textit{Rotated roots of unity:} \quad z_j = e^{i(-\pi + 2\pi j/n)} = -e^{i2\pi j/n}, \quad 0 \le j \le n-1. \tag{3.5}$$

and the (n)-point Chebyshev grid,

$$\textit{Chebyshev points:} \quad x_j = \cos(-\pi + 2\pi j/n), \quad 0 \le j \le n-1, \tag{3.4}$$

Because of the 2-1 correspondence between the θ and the x variables, only $\lfloor n/2 \rfloor + 1$ of these points are distinct.

All three of these settings are unassailably important. Real analysts cannot do without Fourier, complex analysts cannot do without Laurent, and numerical analysts cannot do without Chebyshev. Moreover, the mathematics of the connections between the three frameworks is beautiful. But all this symmetry presents an expository problem. Without a doubt, a fully logical treatment should consider x, z and θ in parallel. Each theorem should appear in three forms. Each application should be one of a trio.

The Chebyshev setting has been dealt with in ATAP. These chapters aim to explore the Fourier case. Here then is the mathematical plan for this book. Our central interest will be the approximation of periodic functions $f(\theta)$ on $[-\pi, \pi]$. When it comes to deriving formulas and proving theorems, however, we shall generally transplant to $F(z)$ on the unit circle so as to make the tools of complex analysis most conveniently available.

Now let us turn to the definitions, already implicit in (3.1)–(3.3). The kth *Fourier mode* e_k can be defined as the function z^k evaluated on the unit circle:

$$\theta = -i \log z = \arccos(x), \quad z = e^{i\theta}, \quad x = \cos\theta. \tag{3.7}$$

$$e_k(\theta) = e^{ik\theta} = z^k = e^{ik \arccos(x)}. \tag{3.8}$$

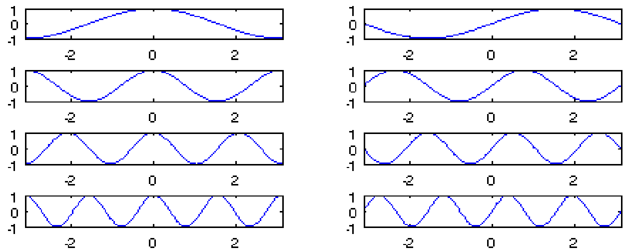
These complex exponentials are a family of orthogonal functions with respect to the complex inner product define on the unit circle, but we shall not make much use of orthogonality until later chapters.

It follows from (3.8) that e_k satisfies $|e_k(\theta)| = 1$ for $\theta \in [-\pi, \pi]$. The real and imaginary parts of e_k are the familiar sin and cos functions. This relationship is perhaps the most remarkable equation of all of mathematics:

$$e^{ik\theta} = (\cos\theta + i\sin\theta)^k = \cos k\theta + i \sin k\theta$$

for any integer k. The first equality above is called *Euler's formula* while the second is called *De Moivre's formula.*

```
FS = 'fontsize';
e = chebfun(@(t) exp(1i*t), [-pi, pi], 'periodic');
einv = chebfun(@(t) exp(-1i*t), [-pi, pi], 'periodic');
for n = 1:4
    subplot(4,2,2*n-1), plot(real(e.^n)), axis([-pi pi -1 1])
    subplot(4,2,2*n), plot(imag(e.^n)), axis([-pi pi -1 1])
end
```



Here are the coefficients of these sinosuoidal functions with respect to the complex exponential basis $\ldots, e^{-i\theta}, 1, e^{i\theta}, \ldots$. As usual, Matlab orders coefficients from highest degree down to degree zero and then we move on the negaitve degrees.

```
for n = 1:3, en = e.^n; fourcoeffs(en).', end
```

```
ans =
  1.000000000000000 - 0.000000000000000i
  0.000000000000000 - 0.000000000000000i
  0.000000000000000 + 0.000000000000000i
ans =
  1.000000000000000 - 0.000000000000000i
  0.000000000000000 - 0.000000000000000i
  0.000000000000001 - 0.000000000000000i
 -0.000000000000000 - 0.000000000000000i
 -0.000000000000000 - 0.000000000000000i
ans =
  1.000000000000000 - 0.000000000000000i
  0.000000000000001 - 0.000000000000000i
  0.000000000000001 - 0.000000000000000i
 -0.000000000000000 + 0.000000000000000i
 -0.000000000000000 - 0.000000000000000i
  0.000000000000000 - 0.000000000000000i
  0.000000000000000 + 0.000000000000000i
```

The $\sin, \cos$ basis is equally familiar and comfortable, and can also be used for numerical work with periodic functions on an interval. If the domain is $[a, b]$ rather than $[-\pi, \pi]$, the frequencies of the Fourier modes must be scaled accordingly, and Chebfun does this automatically when it works on other intervals. For example, $\cos(5\theta)$ has the expansion

$$\cos(5\theta) = \frac{1}{2}e_5(\theta) + i\frac{1}{2}e_{-5}(\theta).$$

We can see this again using the command `fourcoeffs(p)`, where p is the chebfun whose Fourier coefficients we want to know:

```
format short,
c5 = chebfun(@(t) cos(5*t), [-pi, pi], 'periodic');
fourcoeffs(c5).'
```

```
ans =
   0.5000 + 0.0000i
  -0.0000 - 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -0.0000 - 0.0000i
   0.0000 + 0.0000i
  -0.0000 + 0.0000i
   0.0000 - 0.0000i
   0.0000 - 0.0000i
  -0.0000 + 0.0000i
   0.5000 - 0.0000i
```

Any trigonometric polynomial p can be written uniquely as a finite Fourier series: the functions $1, e^{i\theta}, e^{-i\theta}, \ldots, e^{ik\theta}, e^{-ik\theta}$ form a basis for P_k. Since p is determined by its values at Fourier points, it follows that there is a one-to-one linear mapping between values at Fourier points and Fourier expansion coefficients. This mapping can be applied in $O(n \log n)$ operations with the aid of the Fast Fourier Transform (FFT), a crucial observation for practical work that was perhaps first made by Gauss. This is what Chebfun does every time it constructs a periodic chebfun. In fact, Chebfun uses the FFT even when it constructs Chebyshev interplants but we shall not give details.

Just as a trigonometric polynomial p has a finite Fourier series, a more general function f has an infinite Fourier series. Exactly what kind of "more general function" can we allow? For an example like $f(\theta) = e^{\sin(\theta)}$ with a rapidly converging Fourier series, everything will surely be straightforward, but what if f is merely differentiable rather than analytic? Or what if it is continuous but not differentiable? Analysts have studied such cases carefully, identifying exactly what degrees of smoothness correspond to what kinds of convergence of Fourier series. We shall not concern ourselves with trying to state the sharpest possible result but will just make a particular assumption that covers most applications. We shall assume that f is *Lipschitz continuous* on $[-\pi, \pi]$ with $f(-\pi) = f(\pi)$. Recall that this means that there is a constant C such that $|f(\theta_1) - f(\theta_2)| \le C|\theta_1 - \theta_2|$ for all $\theta_1, \theta_2 \in [-\pi, \pi]$. Recall also that a series is *absolutely convergent* if it remains convergent if each term is replaced by its absolute value, and that this implies that one can reorder the terms arbitrarily without changing the result. Such matters are discussed in analysis textbooks such as [Rudin 1976].

Here is our basic theorem about Fourier series and their coefficients.

Theorem 3.1. Fourier series. *If f is Lipschitz continuous on $[-\pi, \pi]$ with $f(-\pi) = f(\pi)$, it has a unique representation as a Fourier series,*

$$f(\theta) = \sum_{k=\infty}^{\infty} a_k e^{ik\theta}, \tag{3.11}$$

which is absolutely and uniformly convergent, and the coefficients are given by the formula

$$a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \tag{3.12}$$

Proof. See any standard text on Fourier analysis.

We now know that any function f, so long as it is Lipschitz continous, has a Fourier series. Chebfun represents a periodic function as a finite series of some degree n, storing both its values at Fourier points and also, equivalently, its Fourier coefficients. How does it figure out the right value of n? Given a set of n samples, it converts the data to a Fourier expansion of degree k, where $n = 2k+1$ and examines the resulting Fourier coefficients. If several of these in a row fall below a relative level of approximately 10^{-15}, then the grid is judged to be fine enough. For example, here are the Fourier coefficients of the chebfun corresponding to $e^{\cos\theta}$:

```
f = chebfun(@(t) exp(cos(t)), [-pi, pi], 'periodic');
format long
a = fourcoeffs(f).'
```

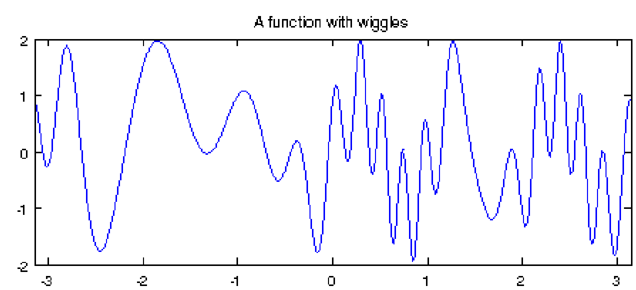
```
a =
  0.000000000000019 + 0.000000000000000i
  0.000000000000520 + 0.000000000000000i
  0.000000000012490 - 0.000000000000000i
  0.000000000275295 + 0.000000000000000i
  0.000000005518386 - 0.000000000000000i
  0.000000099606240 + 0.000000000000000i
  0.000001599218231 - 0.000000000000000i
  0.000022488661477 - 0.000000000000000i
  0.000271463155957 + 0.000000000000000i
  0.002737120221047 + 0.000000000000000i
  0.022168424924332 + 0.000000000000000i
  0.135747669767038 - 0.000000000000000i
  0.565159103992485 - 0.000000000000000i
  1.266065877752008 + 0.000000000000000i
  0.565159103992485 + 0.000000000000000i
  0.135747669767038 + 0.000000000000000i
  0.022168424924332 - 0.000000000000000i
  0.002737120221047 - 0.000000000000000i
```

```
0.000271463155957 - 0.000000000000000i
0.000022488661477 + 0.000000000000000i
0.000001599218231 + 0.000000000000000i
0.000000099606240 + 0.000000000000000i
0.000000005518386 + 0.000000000000000i
0.000000000275295 - 0.000000000000000i
0.000000000012490 + 0.000000000000000i
0.000000000000520 - 0.000000000000000i
0.000000000000019 - 0.000000000000000i
```

Notice that the first and the last coefficients are about at the level of machine precision.

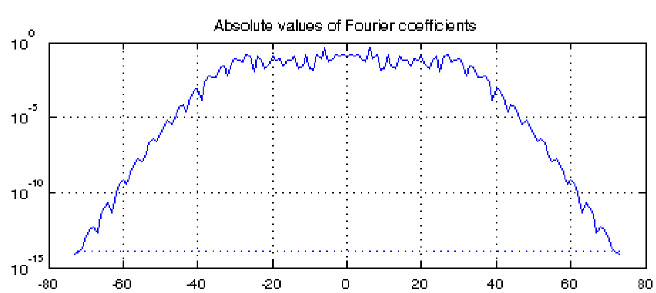
For complicated functions it is often more interesting to plot the coefficients than to list them. For example, here is a function with a number of wiggles:

```
f = chebfun(@(x) sin(6*x) + sin(20*exp(sin(x))), [-pi, pi], 'periodic');
clf, plot(f), title('A function with wiggles',FS,9)
```



If we plot the absolute values of the Fourier coefficients, here is what we find:

```
plotcoeffs(f)
grid on, title('Absolute values of Fourier coefficients',FS,9)
```



One can explain this plot as follows. Up to degree about $k = 30$, a Fourier series cannot resolve f much at all, for the oscillations occur on too short wavelengths. After that, the series begins to converge rapidly. By the time we reach

$k = 70$, the accuracy is about 15 digits, and the computed Chebyshev series is truncated there. We can find out exactly where the truncation took place with the command `length(f)`:

```
length(f)
```

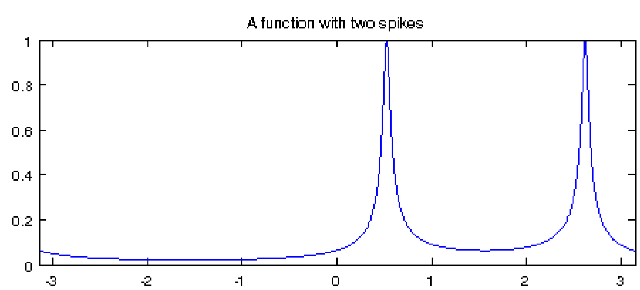
```
ans =
   147
```

This tells us that the chebfun is a trigonometric interpolant through $n = 147$ Fourier points, that is, of degree $k = (n-1)/2 = 73$.

Without giving all the engineering details, here is a fuller description of how Chebfun constructs its approximation. First it calculates the trigonometric interpolant through the function sampled at 9 Fourier points, i.e., a trigonometric polynomial of degree at most 4, and checks whether the Fourier coefficients appear to be small enough. For the example just given, the answer is no. Then it tries 17 points, then 33, then 65, and so on. In this case Chebfun judges at 257 points that the Chebyshev coefficients have fallen to the level of rounding error. At this point it truncates the tail of terms deemed to be negligible, leaving a series of 147 terms. The corresponding degree 73 polynomial is then evaluated at 147 Fourier points via FFT, and these 147 numbers become the data defining this particular chebfun. Engineers would say that the signal has been *downsampled* from 257 points to 147.

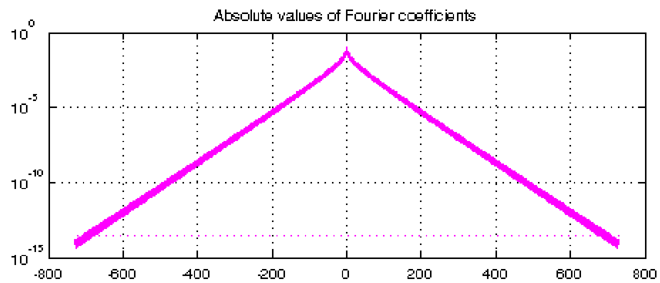
For another example we consider a function with two spikes:

```
f = chebfun(@(x) + 1./sqrt(1+1000*(sin(x)-.5).^2), [-pi, pi], 'periodic');
clf, plot(f), title('A function with two spikes',FS,9)
```



Here are the Fourier coefficients of the chebfun.

```
chebpolyplot(f,'m'), grid on
title('Absolute values of Fourier coefficients',FS,9)
```



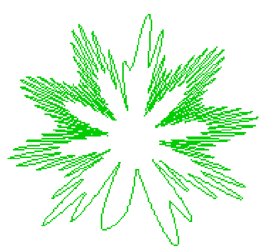
Note that although it is far less wiggly, this function needs ten times as many points to resolve as the previous one. We shall explain the degrees of these trigonometric polynomials in a later chapter.

Fourier interpolants are effective for complex functions (still defined on a real interval) as well as real ones. Here, for example, is a periodic complex function on the interval $[-\pi, \pi]$ and the Fourier representation takes advantage of this fact.

```
f = @(x) (3+sin(10*x)+sin(61*exp(.8*sin(x)+.7))).*exp(1i*x);
p = chebfun(f, [-pi, pi], 'periodic');
```

A plot shows the image of $[-\pi, \pi]$ under f, which appears complicated:

```
plot(p,'linewidth',0.6,'color',[0 .8 0]), axis equal off
```



Yet the degree of the trigonometric polynomial is not so high:

```
(length(p)-1)/2
```

```
ans =
   189
```

People often ask, is there anything special about Fourier points and Fourier polynomials? Could we equally well interpolate in other points and expand in other sets of polynomials? From an approximation point of view, the answer is ? [TODO].

Nevertheless, there is a big advantage of equispaced pionts over non-equispaced points, and this is that one can use the FFT to go from point values to coefficients and back again. There are no known algorithms that make such computations practicable for non-equispaced points.

SUMMARY OF CHAPTER 3. *The Fourier mode* $e^{ik\theta}$ *is an analogue for* $[-\pi, \pi]$ *of the monomial* z^k *on the unit circle. Each periodic Lipschitz continuous function* f *on* $[-\pi, \pi]$ *has an absolutely and uniformly convergent Fourier series, that is, an expansion of the form* $f(\theta) = \sum_{k=-\infty}^{\infty} a_k e^{ik\theta}$.