In "Third Granada Lectures in Computational Physics", Pedro L. Garrido & Joaquín Marro eds., Springer, Berlin, 1995, pages 145-200.

# Fourier Techniques in Numerical Methods for Evolutionary Problems

J.M. Sanz-Serna

Departamento de Matemática Aplicada y Computación,
Facultad de Ciencias, Universidad de Valladolid,
Valladolid, Spain

## 1. Introduction

### 1.1 Scope

These lecture notes contain a summary of the application of Fourier analysis to the numerical solution of time-dependent partial differential equations. The presentation emphasizes two topics: how to use Fourier techniques to analyze and understand finite-difference and finite-element methods and how to derive and code pseudospectral Fourier numerical methods.

The first topic is essential for anyone who wishes to use numerical methods in partial differential equations. While the Von Neumann stability analysis is found in virtually all introductions to the subject, many textbooks do not discuss the ideas of stability and consistency from the Fourier space point of view. Similarly, most elementary texts do not provide an adequate coverage of the notions of numerical dispersion and numerical dissipation. I have tried to fill these gaps.

Pseudospectral methods, our second main topic, are very useful when simulating partial differential equations arising in physics. These methods, being younger than finite-difference/finite-element methods, are not as widely known as they deserve. It is sad that many published papers report finite-difference simulations in situations where a pseudospectral method would have been far more efficient and not more difficult to code.

In order to cater for as wide an audience as possible, virtually no previous knowledge on Fourier analysis or numerical methods has been assumed. The text has been supplemented with exercises and anyone who is really interested in the subject should try and solve most of them. Many exercises contain useful material not covered in the main text. Sometimes the exercises ask for a computer program to be written and MATLAB is an ideal environment for those programs. To help the reader, we have used throughout a set of mathematical conventions that follow those used in MATLAB.

There are eight sections. Sects. 6–8 are devoted to Fourier analysis of finite-difference schemes and Sect. 9 to the pseudospectral method. The first four sections are introductory and serve as a foundation for the last four. Sect. 2 contains a short summary of Fourier series and Sect. 3 deals with partial differential equations. Discrete Fourier analysis, including the Fast Fourier Transform algorithm, is the subject of Sect. 4. I have devoted a full section (Sect. 5) to carefully discussing the relations between Fourier series and discrete Fourier analysis, a point where many expositions are perhaps too terse.

In view of the number of available pages, some important topics have not been covered. Broadly speaking, I only consider problems in one space dimension under periodic solutions. Multidimensional periodic problems can of course be dealt with by Fourier techniques; the adjustments to be made to the material here are minor. On the other hand, problems in the whole space are treated by Fourier integrals rather than by Fourier series; the change from Fourier series to Fourier integrals is sometimes a delicate business (for example, the effects of sampling become more subtle). Homogeneous Dirichlet boundary conditions (vanishing function) or homogeneous Neumann boundary conditions (vanishing normal derivative) on an interval $0 \leq x \leq L$ (or on its multidimensional equivalent $0 \leq x_n \leq L$, where $n$ numbers the space variables) are also amenable to Fourier techniques; they require sine or cosine series, which also have a discrete version with a fast implementation.

### 1.2 Some mathematical preliminaries

It is convenient to list here some basic linear algebra results (Golub and Van Loan 1989; Horn and Johnson 1985) that will be used later. The reader should skip this section now and return to it when referred from other sections.

The *norm* or length of a real or complex vector $\mathbf{X} = (X_1, \ldots, X_\nu)$ with $\nu$ entries is given by $|\mathbf{X}| = (\sum_n |X_n|^2)^{1/2}$. The *norm* of a $\nu \times \nu$ real or complex matrix $A$ is defined by $\|A\| = \max\{|A\mathbf{X}|/|\mathbf{X}| : \mathbf{X} \neq \mathbf{0}\}$. It follows that $|A\mathbf{X}| \leq \|A\|\,|\mathbf{X}|$, for each $\nu \times \nu$ matrix $A$ and $\nu$-vector $\mathbf{X}$, while $\|AB\| \leq \|A\|\|B\|$ if $A$ and $B$ are $\nu \times \nu$ matrices.

If $A$ is a real or complex square matrix, $\rho(A)$, the *spectral radius* of $A$, is the maximum modulus of the eigenvalues of $A$. Since the eigenvalues of a power $A^m$ are the powers of the eigenvalues of $A$, it holds that $\rho(A^m) = \rho(A)^m$. For each square matrix $\rho(A) \leq \|A\|$. On the other hand, $\|A\|$ may be computed by the formula $\|A\| = \rho(A^*A)^{1/2}$, where $A^*$, the *adjoint* of $A$, is the conjugate of the transposed of $A$. The *spectral abscissa*, $\alpha(A)$, of $A$ is the maximum real part of the eigenvalues of $A$.

A *unitary matrix* is a matrix $Q$ for which $Q^* = Q^{-1}$. Unitary matrices preserve vector norms, $|Q\mathbf{X}| = |\mathbf{X}|$, and, by the definition of $\|Q\|$, this implies that $\|Q\| = 1$. Furthermore, unitary matrices preserve matrix norms, in the sense that for unitary $Q$ and arbitrary square $A$, $\|QA\| = \|AQ\| = \|A\|$.

A *normal* matrix is a matrix that commutes with its adjoint $A^*$, i.e, $AA^* = A^*A$. Unitary matrices, real symmetric matrices and real skew-symmetric matrices are obviously normal. A matrix is normal if and only if there exists a unitary matrix $Q$ so that $Q^*AQ$ is the diagonal matrix $\Lambda$ of the eigenvalues $\lambda_n$ of $A$. Since multiplication by unitary matrices does not change matrix norms, for a normal matrix $\|A\| = \|\Lambda\|$ and therefore $\|A\| = \max_n |\lambda_n| = \rho(A)$. From



here, if $A$ is normal, $\|A^m\| = \|A\|^m$. (For nonnormal matrices, $\rho(A) \leq \|A\|$ and $\|A^m\| \leq \|A\|^m$.)

The *exponential* of a square matrix $A$ is, by definition, $\exp(A) = I + A + (1/2)A^2 + \cdots$. It holds that $\exp(A)^{-1} = \exp(-A)$. Also $(d/dt)\exp(tA) = A\exp(tA)$, if $t$ is a real variable. Then $\exp(tA)\mathbf{X}^0$ is the solution of the differential system $(d/dt)\mathbf{X}(t) = A\mathbf{X}(t)$ with initial condition $\mathbf{X}(0) = \mathbf{X}^0$. The eigenvalues of $\exp(A)$ are the numbers $\exp(\lambda)$, with $\lambda$ an eigenvalue of $A$. If $A$ is normal, then $\exp(A)$ is also normal and therefore $\|\exp(A)\| = \rho(\exp(A))$. It then follows trivially (the modulus of the complex exponential is the exponential of the real part) that, if $A$ is normal, then $\|\exp(A)\| = \exp(\alpha(A))$. For nonnormal matrices $\exp(\alpha(A))$ may be smaller than $\|\exp(A)\|$.

If $P(z)$ is a complex polynomial and $A$ is a matrix, the matrix $P(A)$ is defined in an obvious way. For instance, if $P(z) = 3 + 2z + z^2$, then $P(A) = 3I + 2A + A^2$, with $I$ the identity matrix. A *rational function* $R(z)$ of the complex variable $z$ is the quotient of two complex polynomials $R(z) = P(z)/Q(z)$. If $A$ is a square matrix and $R(z)$ is a rational function, then $R(A)$ is, by definition, the matrix $P(A)Q(A)^{-1}$ (or $Q(A)^{-1}P(A)$, because $P(A)$ and $Q(A)^{-1}$ commute). Note that $R(A)$ can only be defined when $Q(A)$ is an invertible matrix. For instance, $R(z) = (1 + z/2)/(1 - z/2)$ is a rational function and $R(A) = [I + (1/2)A][I - (1/2)A]^{-1}$; this is defined if 2 is not amongst the eigenvalues of $A$. The eigenvalues of $R(A)$ are $R(\lambda)$, with $\lambda$ an eigenvalue of $A$. If $A$ is normal, then $R(A)$ is normal and therefore $\|R(A)\| = \rho(R(A))$ is given by the maximum modulus of $R(\lambda)$ as $\lambda$ runs through all the eigenvalues of $A$.

## 2. Review of Fourier Series

### 2.1 The $\mathcal{L}^2$ theory

The literature on Fourier series is of course huge; we only consider the $\mathcal{L}^2$ theory (Strang 1986).

For a given, fixed $L > 0$, we deal with complex-valued functions $f(x)$ of a real variable $x$, $-\infty < x < \infty$, that are $L$-periodic, $f(x) \equiv f(x + L)$. The space $\mathcal{L}^2[0, L]$ consists of all $L$-periodic functions $f$ for which the quantity

$$\|f\| = \left( \int_0^L |f(x)|^2 \, dx \right)^{1/2} \tag{2.1}$$

is finite. For instance, if $x$ represents time and $f$ is the value of a periodic electric current, then the square of the right-hand side of (2.1) is (proportional to) the energy dissipated in a resistor during a period; $\mathcal{L}^2[0, L]$ contains all current functions with finite energy per period.

It is convenient to imagine that each $f$ in $\mathcal{L}^2[0, L]$ is a 'vector' in a space with infinitely many dimensions and that $\|f\|$ is the *norm* or *length* of such a vector. For $f$, $g$ in $\mathcal{L}^2[0, L]$,

$$\langle f, g \rangle = \int_0^L f(x) g(x)^* \, dx$$

($*$ denotes complex conjugate) is the *inner product* of $f$ and $g$. If $\langle f,g\rangle = 0$ we say that $f$ and $g$ are *orthogonal*; we imagine that the vectors $f$ and $g$ are perpendicular.

The system of ($L$-periodic) functions $\ldots,\phi_{-2},\phi_{-1},\phi_0,\phi_1,\phi_2,\ldots$, where

$$\phi_n(x) = \exp\left(\frac{2\pi n i}{L}x\right), \quad n = 0,\pm1,\pm2,\ldots, \tag{2.2}$$

are pairwise orthogonal, $\langle\phi_n,\phi_m\rangle = 0$, $n \neq m$. Just as each geometric vector $\boldsymbol{v}$ in ordinary three-dimensional space can be written in the form

$$\boldsymbol{v} = v_1\boldsymbol{i} + v_2\boldsymbol{j} + v_3\boldsymbol{k} \tag{2.3}$$

in terms of a system $\{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}\}$ of pairwise perpendicular unit vectors, each $f$ in $\mathcal{L}^2[0,L]$ can be referred to the system (2.2):

$$f = \sum_{n=-\infty}^{\infty} \hat{f}_n\phi_n, \tag{2.4}$$

with

$$\hat{f}_n = \frac{1}{L}\langle f,\phi_n\rangle = \frac{1}{L}\int_0^L f(x)\phi_n(x)^*\,dx, \quad n = 0,\pm1,\pm2,\ldots. \tag{2.5}$$

The series in (2.4) is the *Fourier series* of $f$; we use the notation

$$P_N(f) = \sum_{n=-N}^{N} \hat{f}_n\phi_n, \quad N = 0,1,2,\ldots, \tag{2.6}$$

for the corresponding *truncations*. Each term $\hat{f}_n\phi_n$, $n = 0,\pm1,\pm2,\ldots$ in (2.4) is said to be *a Fourier mode* of $f$.

The following properties are crucial.

1. For each $f$ in $\mathcal{L}^2[0,L]$, (2.4) converges to $f$ in the sense that, as $N\to\infty$, $\|f-P_N(f)\|\to 0$. This does not imply that, when the right-hand side of (2.4) is evaluated at a point $x$, the resulting series of complex numbers converges to the value $f(x)$ (pointwise convergence). In the electric current example, the current $f$ is approximated by $P_N(f)$ in such a way that the energy in the residual $f - P_N(f)$ can be made arbitrarily small by taking $N$ suitably large; this does not imply that at a given time $x$ the value of $f(x)$ is the limit, as $N\to\infty$, of the values $(P_N(f))(x)$.
2. For each $f$ in $\mathcal{L}^2[0,L]$, the sequence of Fourier coefficients $\hat{f}_n$ is *square summable*, i.e., $\sum_{-\infty<n<\infty}|\hat{f}_n|^2 < \infty$. Conversely, each square summable sequence of complex numbers corresponds to the Fourier coefficients of a function in $\mathcal{L}^2[0,L]$. Thus the Fourier series defines a correspondence between functions $f$ and sequences of coefficients $\{\hat{f}_n\}$, just as (2.3) provides a correspondence between ordinary vectors $\boldsymbol{v}$ and sets of coefficients $(v_1,v_2,v_3)$: the idea is to use the coefficients $\hat{f}_n$ instead of the function $f$. Furthermore *Parseval's identity* holds:



$$\|f\|^2 = L\sum_{n=-\infty}^{\infty}|\hat{f}_n|^2. \tag{2.7}$$

This is analogous to $|\boldsymbol{v}|^2 = v_1^2+v_2^2+v_3^2$ in (2.3).

3. If $f$ is in $\mathcal{L}^2[0,L]$, then $P_N(f)$ is, among all the linear combinations of the form

$$S_N = \sum_{n=-N}^{N} g_n\phi_n, \tag{2.8}$$

(the $g_n$'s are arbitrary complex numbers) the one that makes $\|f - S_N\|$ as small as possible. Functions of the form (2.8) are called *trigonometric polynomials of degree N*.

**Exercise 1** Prove that the functions in (2.2) are pairwise orthogonal. Prove that each $\phi_n$ has length $\sqrt{L}$, so that the functions $\Phi_n = \phi_n/\sqrt{L}$ are pairwise orthogonal and have unit length. Substitute $\phi_n = \sqrt{L}\Phi_n$ in (2.4) to get $f = \sum\hat{F}_n\Phi_n$, with $\hat{F}_n = \langle f,\Phi_n\rangle$. This is directly analogous to the familiar formulae $v_1 = \langle\boldsymbol{v},\boldsymbol{i}\rangle$, $v_2 = \langle\boldsymbol{v},\boldsymbol{j}\rangle$, $v_3 = \langle\boldsymbol{v},\boldsymbol{k}\rangle$ for the coefficients in (2.3). In terms of the $\hat{F}_n$'s, (2.7) becomes $\|f\|^2 = \sum|\hat{F}_n|^2$, a formula to be compared with $|\boldsymbol{v}|^2 = v_1^2+v_2^2+v_3^2$. Thus the $\Phi_n$'s lead to formulae that are simpler to remember; some authors prefer to use it instead of the $\phi_n$'s in (2.2).

**Exercise 2** Show that, with the definitions in (2.5) and (2.6), the coefficients $\hat{f}_n$ are such that $\langle f,\phi_n\rangle = \langle P_N(f),\phi_n\rangle$, $n = 0,\pm1,\ldots,\pm N$. Denote by $X_N$ the set of trigonometric polynomials (2.8) and prove that $P_N(f)$ is the unique element in $X_N$ for which $f - P_N(f)$ is orthogonal to all functions in $X_N$. Thus $P_N(f)$ is the *orthogonal projection* of $f$ onto $X_N$. This explains Property 3 above.

**Exercise 3** Consider the $L$-periodic square wave function $f(x) = 1$, if $0 < x \le L/2$, $f(x) = -1$, if $L/2 < x \le L$. Show that $\hat{f}_n = -2i/(\pi n)$ for $n$ odd and $\hat{f}_n = 0$ for $n$ even.

**Exercise 4** Consider the $L$-periodic saw-tooth function $f(x) = x$, if $0 < x \le L/2$, $f(x) = L - x$, if $L/2 < x \le L$. Show that $\hat{f}_0 = L/4$, $\hat{f}_n = -L/(\pi n)^2$, for $n$ odd, and $\hat{f}_n = 0$, for $n \neq 0$ even.

## 2.2 The trigonometric version

It is sometimes useful to write $\phi_n$ in trigonometric form

$$\phi_n(x) = \cos\frac{2\pi n}{L}x + i\sin\frac{2\pi n}{L}x;$$

substitution in (2.4) leads to the following trigonometric form of the Fourier series

$$f = c_0(f) + \sum_{n=1}^{\infty} \left( c_n(f) \cos \frac{2\pi n}{L} x + s_n(f) \sin \frac{2\pi n}{L} x \right), \tag{2.9}$$

with

$$c_0(f) = \hat{f}_0,$$
$$c_n(f) = \hat{f}_n + \hat{f}_{-n}, \quad n = 1, 2, \ldots,$$
$$s_n(f) = i(\hat{f}_n - \hat{f}_{-n}), \quad n = 1, 2, \ldots.$$

Using (2.5), one arrives at the following formulae that allow the computation of $c_n(f)$, $s_n(f)$ without using the $\hat{f}_n$'s:

$$c_0(f) = \frac{1}{L} \int_0^L f(x)\, dx, \tag{2.10}$$
$$c_n(f) = \frac{2}{L} \int_0^L f(x) \cos \frac{2\pi n}{L} x\, dx, \quad n = 1, 2, \ldots,$$
$$s_n(f) = \frac{2}{L} \int_0^L f(x) \sin \frac{2\pi n}{L} x\, dx, \quad n = 1, 2, \ldots.$$

The form (2.9) has the advantage that, if $f$ is real-valued, then $c_n(f)$ and $s_n(f)$ are real; the coefficients $\hat{f}_n$ are complex even if $f$ is real.

In (2.9), $f$ is decomposed into a constant $c_0(f)$ (which coincides with the average of $f$ over one period, see (2.10)) and a sum of functions of the form ($c_n = c_n(f)$, $s_n = s_n(f)$)

$$c_n \cos \frac{2\pi n}{L} x + s_n \sin \frac{2\pi n}{L} x, \quad n = 1, 2, \ldots.$$

Assume for a moment that $f$ (and hence $c_n$ and $s_n$) are real. Then

$$c_n \cos \frac{2\pi n}{L} x + s_n \sin \frac{2\pi n}{L} x = A_n \cos \left( \frac{2\pi n}{L} x - \psi_n \right), \tag{2.11}$$

where $A_n = \sqrt{c_n^2 + s_n^2}$, $\psi_n = \arctan(s_n/c_n)$, so that $c_n = A_n \cos\psi_n$ and $s_n = A_n \sin\psi_n$. Thus, the function in (2.11), *the $n$-th harmonic of $f$*, $n = 1, 2, \ldots$, corresponds to a sinusoidal profile, whose amplitude $A_n$ and initial phase $-\psi_n$ are determined by the Fourier coefficients $c_n$ and $s_n$. The (smallest or basic) period of (2.11) as a function of $x$ is $L/n$ so that one period $0 \leq x \leq L$ of $f$ is covered by $n$ cycles of the sinusoid (2.11). Note that in the trigonometric format (2.9), $n$ is nonnegative; both the coefficients $\hat{f}_n$ and $\hat{f}_{-n}$ contribute to the harmonic with period $L/n$, $n = 1, 2, \ldots$. If $f$ is not real-valued this interpretation can be applied to its real and imaginary parts.

The form (2.4) is easier to handle mathematically. The trigonometric form possesses more meaning.

**Exercise 5** Prove that if $f$ is odd, $f(x) \equiv -f(-x)$, then $c_n(f) = 0$, $n = 0, 1, 2, \ldots$. Prove that if $f$ is even, $f(x) \equiv f(-x)$, then $s_n(f) = 0$, $n = 1, 2, \ldots$. Prove the converse of these results.



**Exercise 6** If all you know of $f$ are the coefficients $\hat{f}_n$, how would you tell whether $f$ is real-valued? How would you tell whether $f$ is even, odd?

## 2.3 Fourier series and derivatives

We denote by $\partial_x$ the operator of differentiation with respect to $x$. The system (2.2) is more advantageous than other orthogonal systems because each $\phi_n$ is an *eigenfunction* of the operator $\partial_x$:

$$\partial_x \phi_n = \lambda_n \phi_n, \quad \lambda_n = 2\pi n i/L. \tag{2.12}$$

On $\phi_n$ differentiation reduces to multiplication by the ***eigenvalue*** $\lambda_n$. Hence, from (2.4),

$$\partial_x f = \sum_{n=-\infty}^{\infty} (\lambda_n \hat{f}_n) \phi_n.$$

From Property 2 in Sect. 2.1, we see that $\partial_x f$ exists and belongs to $\mathcal{L}^2[0, L]$ if and only if $\sum |\lambda_n|^2 |\hat{f}_n|^2 < \infty$, i.e., $\sum n^2 |\hat{f}_n|^2 < \infty$. For the square wave in Exercise 3, $n^2 |\hat{f}_n|^2 = 4/\pi^2$ and the series diverges: $f$ has jump discontinuities at $x = 0, \pm L/2, \pm L, \ldots$ and correspondingly $\partial_x f$ has delta functions at those points; $\partial_x f$ is not in $\mathcal{L}^2[0, L]$. For the saw-tooth function in Exercise 4, $n^2 |\hat{f}_n|^2 = 4/(\pi^2 n^2)$, which leads to a convergent series and to $\partial_x f$ in $\mathcal{L}^2[0, L]$. In fact, $\partial_x f$ is the square wave function.

For higher derivatives, $k = 1, 2, \ldots$,

$$\partial_x^k f = \sum_{n=-\infty}^{\infty} (\lambda_n^k \hat{f}_n) \phi_n,$$

and therefore $\partial_x^k f$ exists and belongs to $\mathcal{L}^2[0, L]$ if and only if the Fourier coefficients $\hat{f}_n$ of $f$ decay as $|n| \to \infty$ fast enough for the series $\sum n^{2k} |\hat{f}_n|^2$ to be finite. Smooth functions have Fourier coefficients that decrease fast or, in other words, *the smoother the function the poorer in harmonics with large $n$*. This is intuitive. The function $\phi_n$ varies from $\phi_n = 1$ to $\phi_n = -1$ in an $x$-interval, $0 \leq x \leq L/(2|n|)$, whose length is small for $|n|$ large; therefore large coefficients $|\hat{f}_n|$ for large $|n|$ in (2.4) lead to sharp variations in $f$.

There is another interesting implication. By Parseval's identity (2.7) applied to the function $f - P_N(f)$,

$$\|f - P_N(f)\|^2 = L \sum_{|n|>N} |\hat{f}_n|^2;$$

smooth $f$ corresponds to quickly decreasing $\hat{f}_n$ and hence to small lengths of the residuals $f - P_N(f)$: *the smoother the function, the faster the Fourier series converges*. See Figs. 1 and 2.

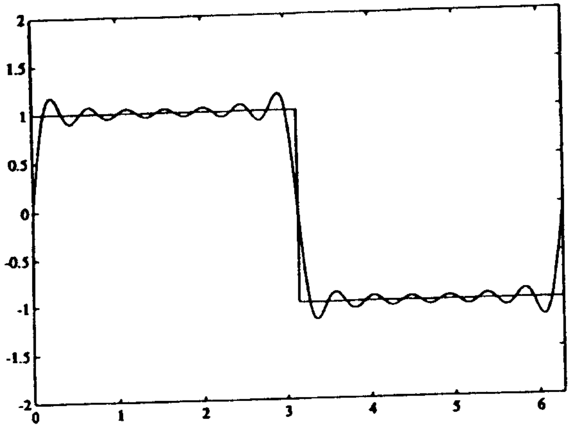


**Fig. 1.** The square wave function (Exercise 3) and the truncation $P_{13}$ of its Fourier series

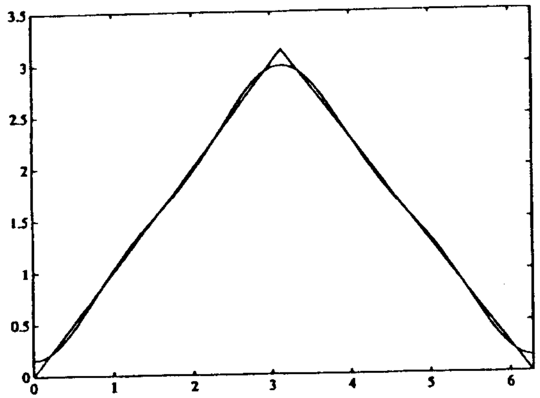


**Fig. 2.** The saw-tooth function (Exercise 4) and the truncation $P_3$ of its Fourier series. Compare with Fig. 1: here fewer harmonics give a better approximation

**Exercise 7** Prove that if $f$ has derivatives of all orders in $\mathcal{L}^2[0, L]$, then its Fourier coefficients decrease faster than any power of $|n|$, i.e., for each $k = 1, 2, \ldots$, $|n|^k |\hat{f}_n| \to 0$ as $|n| \to \infty$.

**Exercise 8** Prove that the Fourier coefficients of (Canuto et al. 1988) $3/(5 - 4\cos x)$ are $2^{-|n|}$ (you may compute the needed integrals by the residue theorem of complex variables). Thus the Fourier coefficients decrease exponentially as $|n| \to \infty$.



**Exercise 9** For the function in Exercise 3, write a program that draws the graphs of $f$ and $P_N(f)$. (Use the trigonometric format to avoid complex quantities.) Run this program for several values of $N$. Do the same for the functions in Exercise 4 and Exercise 8. What are your conclusions?

# 3. Fourier Analysis of Initial Value Problems

## 3.1 Formal solution

Let $P(z) = a_0 + a_1 z + \ldots + a_d z^d$ be a polynomial with complex coefficients. We consider the following periodic initial value problem. We wish to find a complex-valued function $u = u(x,t)$, $-\infty < x < \infty$, $t \geq 0$, $L$-periodic in $x$, that satisfies the differential equation

$$\partial_t u(x,t) = P(\partial_x) u(x,t), \quad -\infty < x < \infty, \quad t > 0, \tag{3.1}$$

along with the initial condition

$$u(x,0) = u^0(x), \quad -\infty < x < \infty, \tag{3.2}$$

where $u^0$ is a given function in $\mathcal{L}^2[0, L]$. For $P(z) = az^2$, $a$ a positive constant, we have $P(\partial_x) = a\partial_x^2$ and (3.1) is the *heat equation* $\partial_t u = a\partial_{xx} u$ with diffusivity constant $a$. The choice $P(z) = -cz$, $c$ real, leads to the *advection equation* $\partial_t u = -c\partial_x u$; for $P(z) = iz^2$, we have the *Schroedinger equation* in nondimensional units $\partial_t u = i\partial_{xx} u$ (or $i\partial_t u = -\partial_{xx} u$), etc.

The problem (3.1)–(3.2) is easily solved by Fourier series (Strang 1986). The $\phi_n$'s are eigenfunctions of $\partial_x$ (see (2.12)) and hence eigenfunctions of the operator $P(\partial_x)$ in (3.1):

$$\begin{aligned} P(\partial_x)\phi_n &= (a_0 + a_1\partial_x + \ldots + a_d \partial_x^d)\phi_n \\ &= (a_0 + a_1\lambda_n + \ldots + a_d \lambda_n^d)\phi_n = P(\lambda_n)\phi_n. \end{aligned}$$

It is convenient to introduce the notation

$$\mu_n = P(\lambda_n)$$

for the eigenvalues of $P(\partial_x)$. For each fixed value of $t$, the solution $u(x,t)$ of (3.1)–(3.2) is sought as a Fourier series

$$u(x,t) = \sum_{n=-\infty}^{\infty} \hat{u}_n(t)\phi_n(x), \tag{3.3}$$

i.e., as a *superposition of eigenfunctions*. Substitution of (3.3) in (3.1) yields

$$\sum_{n=-\infty}^{\infty} \frac{d}{dt}\hat{u}_n(t)\phi_n(x) = \sum_{n=-\infty}^{\infty} \mu_n \hat{u}_n(t)\phi_n(x)$$

or

$$\frac{d}{dt}\hat{u}_n(t) = \mu_n \hat{u}_n(t), \quad n = 0, \pm 1, \pm 2, \dots \tag{3.4}$$

Thus the partial differential equation (3.1) for $u$ is equivalent to the system of infinitely many ordinary differential equations (3.4) for the Fourier coefficients. This system is ***diagonal*** or uncoupled: the $n$-th equation only involves the $n$-th coefficient $\hat{u}_n(t)$. Diagonalization is of course what one looks for when using eigenfunctions. In turn, (3.2) provides the initial values for (3.4),

$$\hat{u}_n(0) = \hat{u}_n^0, \quad n = 0, \pm 1, \pm 2, \dots, \tag{3.5}$$

where $\hat{u}_n^0$ are the Fourier coefficients of $u^0$. From (3.4) and (3.5), $\hat{u}_n(t) = \exp(\mu_n t)\hat{u}_n^0$ and (3.3) reads

$$u(x,t) = \sum_{n=-\infty}^{\infty} \exp(\mu_n t)\hat{u}_n^0 \phi_n(x). \tag{3.6}$$

Each term in this series is called a *mode* of the solution.

**Exercise 10** Particularize (3.6) to the heat, advection and Schroedinger equations mentioned above.

**Exercise 11** Systems of $\nu$ equations with $\nu$ unknown functions can be cast in the format (3.1) by allowing $u$ to be a vector with $\nu$ components and letting the coefficients $a_j$ of $P(z)$ be $\nu \times \nu$ constant matrices ($z$ remains a complex variable). Write $P(z)$ for the system

$$\partial_t v = c\partial_x w, \quad \partial_t w = c\partial_x v, \tag{3.7}$$

($c$ a positive constant). Prove that (3.6) remains valid for systems; $\mu_n = P(\lambda_n)$ and $\exp(\mu_n t)$ are now $\nu \times \nu$ matrices (see Sect. 1.2) and $\hat{u}_n^0$ a vector whose $\nu$ components are the Fourier coefficients of the components of $u^0$. Particularize (3.6) to the system (3.7).

**Exercise 12** Prove that the wave equation $\partial_{tt}\zeta = c^2\partial_{xx}\zeta$ is equivalent to (3.7) through the change of variables $v = \partial_t\zeta$, $w = c\partial_x\zeta$. This illustrates the reformulation of equations involving $\partial_t^k$, $k > 1$, as systems of the first order in $t$ (i.e., systems involving only first derivatives with respect to $t$). The resulting systems may then be solved as in Exercise 11.

### 3.2 Well-posed problems

The solution $u = u(x,t)$ of (3.1)–(3.2) is a function of two arguments. These arguments do not play a symmetric role; it is convenient to introduce the notation $u(\cdot,t)$ to refer to the function of the first argument obtained by giving the fixed numerical value $t$ to the second argument. If for each fixed $t \geq 0$, $u(\cdot,t)$ belongs to $\mathcal{L}^2[0,L]$, then we may imagine that each $u(\cdot,t)$ is a vector in $\mathcal{L}^2[0,L]$; this



vector changes with $t$ starting from the initial vector $u^0$ at $t = 0$. It is this change that we wish to study.

A crucial quantity is given by ($\Re$ denotes real part)

$$\alpha = \alpha(P(\partial_x)) = \sup_{-\infty<n<\infty} \Re\mu_n,$$

the *spectral abscissa* of the operator $P(\partial_x)$ (cf. Section 1.2). Two situations may arise.

(i) $\alpha < \infty$. Then we use (2.7) in (3.6) to get, for $t \geq 0$,

$$\begin{aligned}\|u(\cdot,t)\|^2 &= L\sum_{n=-\infty}^{\infty} |\exp(\mu_n t)|^2 |\hat{u}_n^0|^2 \\ &= L\sum_{n=-\infty}^{\infty} \exp(2\Re\mu_n t)\,|\hat{u}_n^0|^2 \\ &\leq Le^{2\alpha t}\sum_{n=-\infty}^{\infty} |\hat{u}_n^0|^2\end{aligned}$$

or

$$\|u(\cdot,t)\| \leq e^{\alpha t}\|u^0\|. \tag{3.8}$$

Thus, if the initial datum $u^0$ is in $\mathcal{L}^2[0,L]$, the solution $u(\cdot,t)$ remains in $\mathcal{L}^2[0,L]$ at all later times $t > 0$. Furthermore, for each fixed $t > 0$, the length of the evolved vector $u(\cdot,t)$ can be bounded by a factor $e^{\alpha t}$ (independent of $u^0$) times the initial length $\|u^0\|$. Small $u^0$ lead to small solutions. The problem is said to be *well posed* (Richtmyer and Morton 1967; Kreiss and Oliger 1973; Sanz-Serna 1985; Sanz-Serna and Verwer 1989).

(ii) $\alpha = \infty$. Then, for the initial condition $u^0 = \phi_n$, with norm $\sqrt{L}$, the solution $u(\cdot,t) = \exp(\mu_n t)\phi_n$ has a length $\sqrt{L}\exp(\Re\mu_n t)$ that can be made arbitrarily large by varying $n$. It is therefore impossible to bound $\|u(\cdot,t)\|$ by a $u^0$-independent factor times $\|u^0\|$. The problem is said to be *ill posed*. Initial conditions close to 0 may result in arbitrarily large solutions. Such problems are not good candidates to become physical models.

**Exercise 13** Prove that the equations in Exercise 10 give rise to well-posed initial value problems. How about the *backward heat equation* $\partial_t u = -a\partial_{xx}u$, $a$ a positive constant?

**Exercise 14** A system (cf. Exercise 11) leads to a well-posed initial value problem if there exist constants $K$ and $\alpha$ such that

$$\sup_{-\infty<n<\infty} \|\exp(\mu_n t)\| \leq Ke^{\alpha t}, \quad t > 0.$$

(Here $\|\cdot\|$ denotes norm for $\nu \times \nu$ matrices, see Sect. 1.2.) For well-posed systems, derive an estimate similar to (3.8). Study the well-posedness of the initial value problem for (3.7).

### 3.3 Dissipation and dispersion

In this subsection we assume for simplicity that the initial datum $u^0$ in (3.2) is real-valued and write it in trigonometric form as in (2.11), i.e.,

$$u^0(x) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2\pi n}{L}x - \psi_n\right).$$

We also assume that in (3.1) the variables $x$ and $t$ correspond to physical space and time.

Let us consider first the advection equation $\partial_t u = -c\partial_x u$ ($c$ a real constant). The solution (3.6) written in trigonometric form is

$$u(x,t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2\pi n}{L}x - \frac{2\pi n}{L}ct - \psi_n\right). \qquad (3.9)$$

Therefore $u$ is a superposition of sinusoidal waves (Whitham 1974)

$$A_n \cos\left(\frac{2\pi n}{L}x - \frac{2\pi n}{L}ct - \psi_n\right); \qquad (3.10)$$

each of these is constant on the lines in space-time with equations $x - ct = \xi$, $\xi$ a constant. In other words, (3.10) propagates with velocity $c$ without changing shape. Since this holds for each $n$, the same is true for the sum in (3.9) and in fact, $u(x,t) = u^0(x - ct)$, see Figs. 3–4.

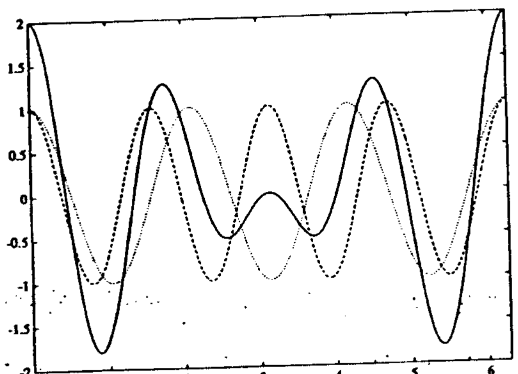


**Fig. 3.** Initial condition (solid line) $u^0(x) = \cos 3x + \cos 4x$ ($L = 2\pi$) obtained by superposing the wave numbers 3 (dotted line) and 4 (dashed line)

In (3.10), $\kappa_n = 2\pi n/L$ is the *wave number*, i.e., the number of complete cycles of the cosine function per $2\pi$ units of length, while $\omega_n = 2\pi nc/L$ is the *(angular) frequency*, i.e., the number of cosine cycles per $2\pi$ units of time. The quotient $\omega_n/\kappa_n = c$ provides the (phase) velocity of the wave. On the other



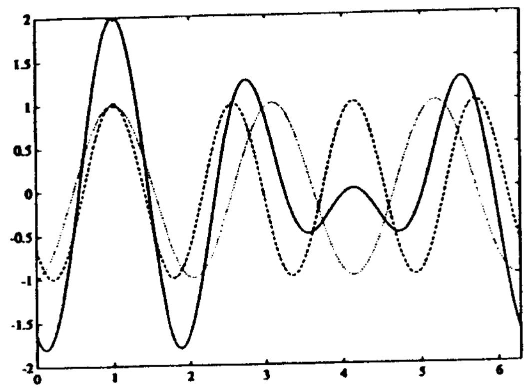


**Fig. 4.** Solution $u(\cdot,t)$ at time $t = 1$ of the advection equation $\partial_t u = -c\partial_x u$, $c = 1$, for the initial condition in Fig. 3. Both harmonics have travelled one unit to the right

hand, $\ell_n = L/n$ is the *wave length*, (distance between two consecutive maxima of the cosine function) and $T_n = L/(cn)$ is the period in time of the wave. Again $\ell_n/T_n = c$.

Let us now turn to the equation $\partial_t u = \partial_{xxx} u$ with solution

$$u(x,t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2\pi n}{L}x - \left[\frac{2\pi n}{L}\right]^3 t - \psi_n\right).$$

We are again superposing sinusoidal waves. But now the wave with wave number $\kappa_n = 2\pi n/L$ travels with a velocity $(2\pi n/L)^2$ that is not the same for all wave numbers. This behavior is known as *dispersion*. When dispersion is present the 'shape' of $u(\cdot,t)$ changes with $t$. This is illustrated in Fig. 5.

The equation $\omega_n = \kappa_n^3$ relating frequency and wave number is known as the *dispersion relation* of the equation (Whitham 1974).

Finally for the heat equation $\partial_t = a\partial_{xx} u$, $a$ positive, the solution is

$$u(x,t) = A_0 + \sum_{n=1}^{\infty}\left[A_n \exp\left(-\frac{4\pi^2 a n^2}{L^2}t\right)\right]\cos\left(\frac{2\pi n}{L}x - \psi_n\right);$$

the sinusoidal components do not move with time; this is not a wave-like equation. It is the amplitude of the components that changes with $t$. The harmonics decay, which physically would correspond to a *dissipative* behavior. Higher wave numbers decay at a faster rate (see Fig. 6). A large diffusivity constant $a$ results in a faster decay.

**Exercise 15** Write in trigonometric form the solutions of the initial value problem for the equation $\partial_t u = \partial_{xx} u + \partial_{xxx} u$. Study the well-posedness of the initial

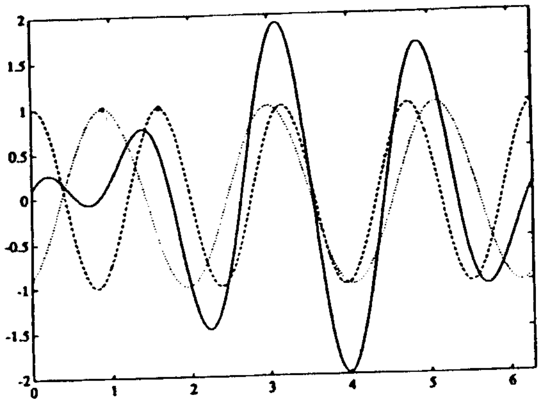


Fig. 5. Solution $u(\cdot,t)$ at time $t = 0.1$ of the equation $\partial_t u = \partial_{xxx} u$ for the initial condition in Fig. 3. The wave number 3 moves with velocity 9, so that the maximum that was initially at $x = 0$ is now at $x = 0.9$. The wave number 4 has velocity 16 and the maximum initially at $x = 0$ is now at $x = 1.6$. (These maxima are indicated by small circles.) Clearly the shape of the solution $u(\cdot,t)$ changes with t

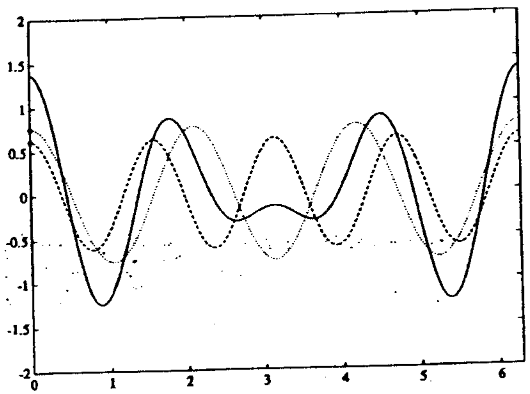


Fig. 6. Solution $u(\cdot,t)$, $t = 0.03$ of the heat equation $\partial_t = \partial_{xx} u$, for the initial condition in Fig. 3

value problem and discuss the solution behavior, that combines dissipation and dispersion.

**Exercise 16** Write in trigonometric form the solutions of the initial value problem for the equation $\partial_t u = \partial_{xx} u + au$, $a$ a real constant. Study the well-posedness of the initial value problem. Discuss the solution behavior for different values of $a$. Note that solution growth is compatible with well-posedness.



# 4. Discrete Fourier Analysis

## 4.1 The discrete Fourier transform

The discrete Fourier transform (Strang 1986) is one of the main tools of modern applied mathematics. For the time being, it is convenient not to think at all of the discrete transform as a discrete version of the Fourier series; the relation between discrete transforms and Fourier series is discussed in Sect. 5 below.

We work in the space $\mathbb{C}^M$ of column vectors $\mathbf{X}$ with $M$ complex components $\mathbf{X} = [X_0, X_1, \dots, X_{M-1}]^T$; note that subscripts run from 0 to $M-1$, rather than the standard 1 to $M$. The superscript $T$ means transpose.

The ($M$-dimensional) ***discrete Fourier transform*** is the linear transformation in $\mathbb{C}^M$ that associates with each vector $\mathbf{X}$ the vector $F_M\mathbf{X}$, where $F_M$ is the $M \times M$ complex matrix whose entry $(\ell, n)$, $\ell, n = 0, 1, \dots, M-1$, is $w_M^{\ell n}$, $\ell n$-th power of the number

$$w_M = \exp\left(-\frac{2\pi i}{M}\right) = \cos\frac{2\pi}{M} - i\sin\frac{2\pi}{M}. \tag{4.1}$$

For instance, for $M = 2$, $w_2 = -1$ and

$$F_2 = \begin{bmatrix} w_2^0 & w_2^0 \\ w_2^0 & w_2^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix};$$

for $M = 3$, $w_3 = -1/2 - i\sqrt{3}/2$ and

$$F_3 = \begin{bmatrix} w_3^0 & w_3^0 & w_3^0 \\ w_3^0 & w_3^1 & w_3^2 \\ w_3^0 & w_3^2 & w_3^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1/2 - i\sqrt{3}/2 & -1/2 + i\sqrt{3}/2 \\ 1 & -1/2 + i\sqrt{3}/2 & -1/2 - i\sqrt{3}/2 \end{bmatrix};$$

for $M = 4$, $w_4 = -i$ and

$$F_4 = \begin{bmatrix} w_4^0 & w_4^0 & w_4^0 & w_4^0 \\ w_4^0 & w_4^1 & w_4^2 & w_4^3 \\ w_4^0 & w_4^2 & w_4^4 & w_4^6 \\ w_4^0 & w_4^3 & w_4^6 & w_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}.$$

It is important to observe that $w_M^M = 1$, i.e., $w_M$ is an $M$-th root of 1. In fact, $w_M$ is the $M$-th root of unity whose argument is negative and as small as possible.

The idea behind the use of the discrete transform is that many vector operations are easier if performed on the transformed vectors. One then (i) transforms the data $\mathbf{X}$, (ii) operates with the transformed $F_M\mathbf{X}$ to find the transformed of the solution and (iii) transforms back to find the solution. The third step requires the knowledge of the inverse matrix $F_M^{-1}$, which is really simple:

$$F_M^{-1} = \frac{1}{M}\bar{F}_M. \tag{4.2}$$

Here $\bar{F}_M$ is the matrix obtained by conjugating all the entries of $F_M$. Since $F_M$ and $F_M^{-1}$ only differ in the factor $M$ and in the change $i \to -i$, the algorithms used to compute discrete transforms (see Sect. 4.3) are easily adapted to compute inverse discrete transforms.

On the other hand, $F_M = F_M^T$ and hence (see Sect. 1.2) $F_M^* = \bar{F}_M = MF_M^{-1}$. If $\mathbf{Y} = F_M\mathbf{X}$, then

$$|\mathbf{Y}|^2 = \mathbf{Y}^*\mathbf{Y} = (F_M\mathbf{X})^*\mathbf{Y} = M\mathbf{X}^*F_M^{-1}F_M\mathbf{X} = M|\mathbf{X}|^2,$$

i.e.,

$$\sum_{n=0}^{M-1}|X_n|^2 = \frac{1}{M}\sum_{n=0}^{M-1}|Y_n|^2; \tag{4.3}$$

this is a discrete version of Parseval's identity (2.7). Except for the normalizing factor $M$, $F_M$ is a unitary matrix and using $F_M\mathbf{X}$ instead of $\mathbf{X}$ does not alter the length of the vectors involved.

**Exercise 17** Write explicitly $F_6$ and $F_8$. Use (4.2) to write explicitly $\bar{F}_M$ and $F_M^{-1}$, $M = 2,3,4,6,8$. Have you observed that $\bar{F}_M$ may be obtained by permuting the columns of $F_M$? Check that $F_4F_4^{-1}$ yields the unit matrix.

**Exercise 18** Two vectors $\mathbf{X}_1$, $\mathbf{X}_2$ in $\mathbb{C}^M$ are said to be orthogonal if their inner product $\mathbf{X}_1^*\mathbf{X}_2$ vanishes. Show that the column vectors of $F_M$ are pairwise orthogonal. Show that the same is true for the column vectors of $F_M^{-1}$.

**Exercise 19** Prove that $F_M\bar{F}_M = MI$; this yields (4.2).

## 4.2 An application: systems of ordinary differential equations with circulant matrices

Many situations give rise to systems of the form

$$\frac{d}{dt}\mathbf{X}(t) = A\mathbf{X}(t), \tag{4.4}$$

with $A$ a *circulant* constant complex matrix (Strang 1986)

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{M-1} \\ a_{M-1} & a_0 & a_1 & \cdots & a_{M-2} \\ a_{M-2} & a_{M-1} & a_0 & \cdots & a_{M-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_0 \end{bmatrix}$$

It is fortunate that, for $n = 0,1,\dots,M-1$, the $n$-th column of $F_M^{-1}$ (see (4.2))

$$\mathbf{V}_n = (1/M)\begin{bmatrix} \bar{w}_M^0 \\ \bar{w}_M^n \\ \bar{w}_M^{2n} \\ \vdots \\ \bar{w}_M^{(M-1)n} \end{bmatrix} = (1/M)\begin{bmatrix} \exp\left(\frac{2\pi 0ni}{M}\right) \\ \exp\left(\frac{2\pi ni}{M}\right) \\ \exp\left(\frac{2\pi 2ni}{M}\right) \\ \vdots \\ \exp\left(\frac{2\pi(M-1)ni}{M}\right) \end{bmatrix} \tag{4.5}$$



is an eigenvector of $A$. More precisely

$$A\mathbf{V}_n = \sigma_n\mathbf{V}_n, \quad n = 0,1,\dots,M-1, \tag{4.6}$$

with

$$\sigma_n = a_0\bar{w}_M^0 + a_1\bar{w}_M^n + \dots + a_{M-1}\bar{w}_M^{(M-1)n}, \quad \bar{w}_M = \exp\left(\frac{2\pi i}{M}\right).$$

We therefore look for the solution $\mathbf{X}(t)$ of (4.4) as a *superposition of eigenvectors*

$$\mathbf{X}(t) = \sum_{n=0}^{M-1} Y_n(t)\mathbf{V}_n, \tag{4.7}$$

(the $Y_n(t)$ are complex numbers) or in matrix notation (a product matrix-times-vector is the linear combination of the matrix columns whose coefficients are the components of the vector)

$$\mathbf{X}(t) = F_M^{-1}\mathbf{Y}(t). \tag{4.8}$$

The new vector $\mathbf{Y} = F_M\mathbf{X}$ is therefore the transform of $\mathbf{X}$. Substitution of (4.7) in (4.4) leads, in view of (4.6), to

$$\sum_{n=0}^{M-1}\frac{d}{dt}Y_n(t)\mathbf{V}_n = \sum_{n=0}^{M-1}\sigma_n Y_n(t)\mathbf{V}_n,$$

i.e.,

$$\frac{d}{dt}Y_n(t) = \sigma_n Y_n(t), \quad n = 0,1,\dots,M-1; \tag{4.9}$$

in terms of the $Y_n$'s the system has uncoupled or diagonalized and is easily integrated:

$$Y_n(t) = \exp(\sigma_n t)Y_n(0), \quad n = 0,1,\dots,M-1. \tag{4.10}$$

We conclude from (4.7) and (4.9) that the solution is

$$\mathbf{X}(t) = \sum_{n=0}^{M-1}\exp(\sigma_n t)Y_n(0)\mathbf{V}_n. \tag{4.11}$$

In practice, to find $\mathbf{X}(t)$ at any given numerical value of $t$, one computes $\mathbf{Y}(0)$ by a discrete transform (see (4.8)), integrates in the $Y_n$ variables as in (4.10) and returns to the $X_n$ variables by performing the inverse discrete transform in (4.8) at time $t$.

The reader has certainly noticed the similarity between this material and the contents of Sect. 3.1; $u(x,t)$ corresponds to $\mathbf{X}(t)$, $P(\partial_x)$ to $A$, the $\hat{u}_n$'s to the $Y_n$'s, (3.1) corresponds to (4.4), (3.3) to (4.7), (3.4) to (4.9) and (3.6) to (4.11). At each fixed value of $t$, $u(x,t)$ is parameterized by a continuum of values of $x$; here $\mathbf{X}(t)$ is parametrized by the discrete subscript $n = 0,1,\dots,M-1$. There is a Fourier coefficient $\hat{u}_n$ for each integer $n$, however there are only $M$ variables $Y_n$. The basis functions $\phi_n$ in (3.3) are pairwise orthogonal and so are the vectors $\mathbf{V}_n$ in (4.7), see Exercise 18.

Other applications of the discrete Fourier transform involve the solution of algebraic systems of linear equations, see Exercise 21, and the computation of convolutions (Strang 1986).

**Exercise 20** Prove (4.6).

**Exercise 21** Solve the linear algebraic equations $A\mathbf{X} = \mathbf{B}$, where $A$ is the matrix in (4.4) and $\mathbf{B}$ a known vector. (Hint: If $\mathbf{X} = F_M^{-1}\mathbf{Y}$ and $\mathbf{B} = F_M^{-1}\mathbf{C}$, then $Y_n = C_n/\sigma_n$, $n = 0, 1, \ldots, M-1$.) Prove that to solve such a system one needs one discrete transform, one inverse discrete transform and $M$ divisions. This idea can be extended to more general matrices (Strang 1986).

## 4.3 The fast Fourier transform

### 4.3.1 Preliminary remarks

Once $F_M$ has been formed as in Sect. 4.1, to find $F_M\mathbf{X}$ for a given vector $\mathbf{X}$ requires $M^2$ complex multiplications if one follows the standard recipe for matrix/vector products (each entry in $F_M$ has to be multiplied by an element of $\mathbf{X}$). In 1965 Cooley and Tukey popularized an algorithm, ***the Fast Fourier Transform, FFT,*** that finds $F_M\mathbf{X}$ with less than $(1/2)M\log_2 M$ multiplications (the exact number depends on the details of the specific implementation used). This implies enormous savings. For $M = 2^{12} = 4096$, a dimension that is typical in many applications, $M^2 = 2^{24}$, and the FFT requires less than $6 \times 2^{12}$ multiplications; this means that FFT is at least 600 times faster. Since $\log_2 M$ grows very slowly with $M$, the cost of the FFT grows for all practical purposes like $O(M)$; the straightforward matrix-times-vector algorithm has an $O(M^2)$ cost.

The idea behind the FFT is not difficult (Strang 1986). Suppose that $M$ is even $M = 2N$ and we need to compute $\mathbf{Y} = F_M\mathbf{X}$. We begin by splitting $\mathbf{X}$ into two vectors of length $N$

$$\mathbf{X}' = [X_0, X_2, \ldots, X_{M-2}]^T, \quad \mathbf{X}'' = [X_1, X_3, \ldots, X_{M-1}]^T$$

and computing two transforms

$$\mathbf{Y}' = F_N\mathbf{X}', \quad \mathbf{Y}'' = F_N\mathbf{X}'', \tag{4.12}$$

whose dimension is only *one half* of that of the sought transform. We may recover $\mathbf{Y}$ from $\mathbf{Y}'$ and $\mathbf{Y}''$. In fact, it is straightforward to prove that the first $N$ components of $\mathbf{Y}$ are given by

$$Y_n = Y_n' + w_M^n Y_n'', \quad n = 0, 1, \ldots, N-1, \tag{4.13a}$$

while the $N$ last components are given by

$$Y_{N+n} = Y_n' - w_M^n Y_n'', \quad n = 0, 1, \ldots, N-1. \tag{4.13b}$$



Note that (4.13a)–(4.13b) only require $N$ multiplications $w_M^n Y_n''$. This evaluation of $M$-dimensional transforms in terms of $N$-dimensional transforms is the essence of the FFT.

Even though in practice one implements directly (4.13), the idea above is more easily grasped by rewriting (4.13) in matrix form. For instance, with $M = 4$, $N = 2$, we have

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = M_4 \begin{bmatrix} Y_0' \\ Y_1' \\ Y_0'' \\ Y_1'' \end{bmatrix},$$

where

$$M_4 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -i \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & i \end{bmatrix}.$$

Therefore, by (4.12), ($O_2$ denotes the $2 \times 2$ zero matrix)

$$\mathbf{Y} = M_4 \begin{bmatrix} F_2\mathbf{X}' \\ F_2\mathbf{X}'' \end{bmatrix} = M_4 \begin{bmatrix} F_2 & O_2 \\ O_2 & F_2 \end{bmatrix} \begin{bmatrix} \mathbf{X}' \\ \mathbf{X}'' \end{bmatrix}$$

$$= M_4 \begin{bmatrix} F_2 & O_2 \\ O_2 & F_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X};$$

the rightmost matrix, that we shall denote by $P_4$, permutes the entries of $\mathbf{X} = [X_0, X_1, X_2, X_3]^T$ to give $P_4\mathbf{X} = [X_0, X_2, X_1, X_3]^T$. To sum up

$$F_4 = M_4 \begin{bmatrix} F_2 & O_2 \\ O_2 & F_2 \end{bmatrix} P_4.$$

In the case $M = 8$, $N = 4$, the formulae (4.13) lead in the same way to

$$F_8 = M_8 \begin{bmatrix} F_4 & O_4 \\ O_4 & F_4 \end{bmatrix} P_8,$$

where

$$M_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & w_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & w_8^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w_8^3 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -w_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -w_8^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -w_8^3 \end{bmatrix}$$

and $P_8$ is the permutation matrix

$$P_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

### 4.3.2 The algorithm

Assume now that $M$ is a power of 2, $M = 2^\mu$, $\mu$ a positive integer. The idea just described can be successively applied to reduce the computation of transforms of dimension $M$ to transforms of dimension $M/2$, $M/4$, $M/8$, etc. One goes all the way down to 2-dimensional transforms that are, of course, trivially computed. (A 2-transform requires just two additions.)

For instance, on combining the examples of Sect. 4.3.1, we get for $M = 8$

$$\begin{aligned} \mathbf{Y} = F_8\mathbf{X} &= M_8 \begin{bmatrix} F_4 & O_4 \\ O_4 & F_4 \end{bmatrix} P_8\mathbf{X} \\ &= M_8 \begin{bmatrix} M_4 \begin{bmatrix} F_2 & O_2 \\ O_2 & F_2 \end{bmatrix} P_4 & O_4 \\ O_4 & M_4 \begin{bmatrix} F_2 & O_2 \\ O_2 & F_2 \end{bmatrix} P_4 \end{bmatrix} P_8\mathbf{X} \\ &= M_8 \begin{bmatrix} M_4 & O_4 \\ O_4 & M_4 \end{bmatrix} \begin{bmatrix} F_2 & O_2 & O_2 & O_2 \\ O_2 & F_2 & O_2 & O_2 \\ O_2 & O_2 & F_2 & O_2 \\ O_2 & O_2 & O_2 & F_2 \end{bmatrix} \begin{bmatrix} P_4 & O_4 \\ O_4 & P_4 \end{bmatrix} P_8\mathbf{X}; \end{aligned}$$

to find $\mathbf{Y}$ we successively 'multiply' $\mathbf{X}$ by $P_8, \dots, M_8$. The first two products require only permutations of the entries of $\mathbf{X}$; the last two products implement formulae (4.13) to find two 4-transforms and the sought 8-transform.

For other values of $M = 2^\mu$, the algorithm works in the same way. The matrix $F_M$ is thought of as a product of $(\mu - 1) + 1 + (\mu - 1)$ factors. The rightmost $\mu - 1$ factors describe permutations of the entries of $\mathbf{X}$. The central factor is the multiplication-free computation of 2-transforms. The left-most $\mu - 1$ factors successively yield $2^2$-transforms, $2^4$-transforms, etc.

### 4.3.3 Practical issues

The FFT also works when $M$ is not a power of 2 (Cooley and Tukey 1965). For instance if $M$ has a prime factor $M_1 \neq 2$, $M = M_1 M_2$, one can work out formulae similar to (4.13) to reduce the computation of an $M$-transform to that of $M_1$ transforms of dimension $M_2$. Iteration of this idea reduces the computation of transforms of arbitrary length to that of transforms whose size is a prime factor



2, 3, 5, .... However, in most implementations, the FFT is most efficient when $M$ is of the form $2^\mu$ and one should try to use numbers of this form.

Note that $F_M\mathbf{X}$ is a complex vector even if, as it is often the case in the applications, $\mathbf{X}$ has real entries. The use of the fully complex FFT algorithm described above is not optimal in terms of efficiency and improvements exist (Press et al. 1989).

All good mathematical software libraries possess implementations of the FFT. These should always be preferred to 'home made' implementations written by users. There is a lack of uniformity in the literature when it comes to defining the matrices $F_M$ and $F_M^{-1}$; the definition used here has $-i$ in $F_M$ (see (4.1)) and hence $+i$ in $F_M^{-1}$ (see (4.2)), other authors take the signs differently. Here $F_M^{-1}$ carries the factor $M^{-1}$; other authors attach this factor to the direct transform $F_M$. These variations should cause no problem.

Our definitions coincide with those in MATLAB. The MATLAB function **fft(X)** finds the discrete transform of the vector $\mathbf{X}$ and **ifft(X)** provides the inverse discrete transform. We note that, while mathematical notation has the components of $\mathbf{X}$ labelled $0, 1, \dots, M-1$, components in MATLAB run from 1 to $M$; the mathematical value $X_n$ should then be invoked in a program as $X(n+1)$.

**Exercise 22** Write an FFT program. You can check what you write against a coding from a textbook (Press et al. 1989).

**Exercise 23** Prove that if the linear system in Exercise 21 is solved by discrete Fourier transforms, the overall number of required multiplications and divisions is less than $M \log_2 M + M$. This should be compared with the $O(M^3)$ cost when using Gaussian elimination (Golub and Van Loan 1989). Since, for realistic values of $M$, $\log_2 M$ is a small number, the cost of ***computing*** the solution by Fourier techniques is $O(M)$ for all practical purposes. The cost of ***printing*** the solution at the end of the computation also grows like $O(M)$!

## 5. Discrete Fourier Transform vs. Fourier Series

### 5.1 A first look at aliasing

So far Fourier series, dealing with $L$-periodic functions of $x$, and Fourier transforms, dealing with $M$-vectors, have been presented as unrelated entities. This must now stop.

Let us again consider $L$-periodic complex-valued functions $f$ as in Sects. 2–3. If $M \geq 1$ is an integer, we discretize the variable $x$ by introducing the grid points $x_n = n\Delta x$, $n = 0, \pm 1, \pm 2, \dots$, $\Delta x = L/M$. In applications where the variable $x$ corresponds to physical time, we can think of the grid values $f(x_n)$ as *stroboscopic* samples of $f$. Due to periodicity, $f(x_n) = f(x_m)$ if $n - m$ differ by an integer multiple of $M$. Hence only the points $x_0, x_1, \dots, x_{M-1}$ carry independent information. To each $L$-periodic function $f$ we associate a vector $\mathbf{X}(f)$ in $\mathbb{C}^M$

defined by $[f(x_0), f(x_1), \dots, f(x_{M-1})]^T$. Note that $\mathbf{X}(f)$ depends on $M$, i.e., on the particular grid chosen; for simplicity this dependence is not incorporated into the notation.

Two different functions $f_1$, $f_2$ may have $\mathbf{X}(f_1) = \mathbf{X}(f_2)$; this happens if and only if $f_1$ and $f_2$ coincide at all grid points. A prime example is given by the Fourier basis functions $\phi_n$ in (2.2). For these,

$$\mathbf{X}(\phi_n) = \begin{bmatrix} \exp\left(\frac{2\pi 0 n i}{M}\right) \\ \exp\left(\frac{2\pi n i}{M}\right) \\ \exp\left(\frac{2\pi 2 n i}{M}\right) \\ \vdots \\ \exp\left(\frac{2\pi (M-1) n i}{M}\right) \end{bmatrix} \tag{5.1}$$

and it is easy to check that

$$\mathbf{X}(\phi_n) = \mathbf{X}(\phi_m) \quad \Leftrightarrow \quad n \equiv m, \tag{5.2}$$

where $n \equiv m$ means that $n$ and $m$ differ in an integer multiple of $M$. Hence, *on the grid,* only $\phi_0$, $\phi_1$, ..., $\phi_{M-1}$ are different; $\phi_M$ coincides with $\phi_0$. $\phi_{M+1}$ coincides with $\phi_1$, etc. Also, $\phi_{-1}$ coincides with $\phi_{M-1}$; $\phi_{-2}$ coincides with $\phi_{M-2}$, etc. This coincidence is called *aliasing* and plays a crucial role in discrete Fourier analysis; when $\mathbf{X}(\phi_m) = \mathbf{X}(\phi_n)$ we say that $\phi_m$ is an alias of $\phi_n$ (see Fig. 7).

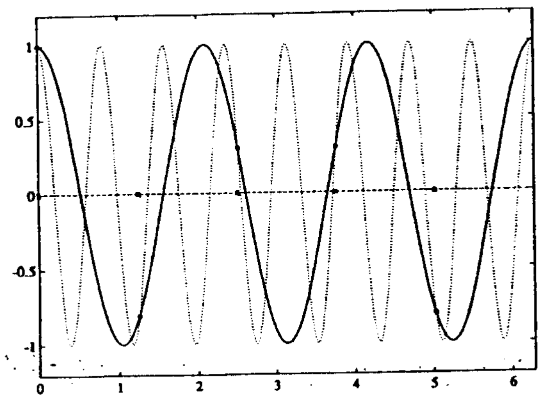


Fig. 7. On the grid resulting from dividing the interval $[0, 2\pi]$ into $M = 5$ equal parts ($\times$ signs) the function $\exp(8ix)$ is an alias of $\exp(3ix)$. This figure corresponds to the real part; a similar figure may be drawn for the imaginary part

**Exercise 24** Check (5.1) and (5.2).



## 5.2 Trigonometric interpolation

### 5.2.1 The bad way

Let us now compare (4.5) and (5.1):

$$\mathbf{V}_n = (1/M)\mathbf{X}(\phi_n), \quad n = 0, 1, \dots, M-1. \tag{5.3}$$

This is an important relation between Fourier transforms and Fourier series: the grid values $\mathbf{X}(\phi_n)$ of the Fourier series basis functions $\phi_n$ coincide, except for the factor $M$, with the columns $\mathbf{V}_n$ of the matrix $F_M^{-1}$. We use this coincidence to solve an interpolation problem. Assume that $f$ is a given function and we wish to find coefficients $a_0, a_1, \dots, a_{M-1}$ such that the trigonometric polynomial

$$\sum_{n=0}^{M-1} a_n \phi_n(x) \tag{5.4}$$

matches $f$ at the grid points, i.e.,

$$f(x_m) = \sum_{n=0}^{M-1} a_n \phi_n(x_m), \quad m = 0, \pm 1, \pm 2, \dots.$$

This condition may be rewritten as

$$\mathbf{X}(f) = \sum_{n=0}^{M-1} a_n \mathbf{X}(\phi_n)$$

or, in view of (5.2) (see the remark before (4.8)),

$$\mathbf{X}(f) = M \sum_{n=0}^{M-1} a_n \mathbf{V}_n = M F_M^{-1} \mathbf{a}.$$

Therefore

$$\mathbf{a} = (1/M) F_M \mathbf{X}(f), \tag{5.5}$$

the $a_n$ are $1/M$ times the entries of the transform $F_M \mathbf{X}(f)$ of the grid values $\mathbf{X}(f)$ of $f$.

The trouble with this interpolation is that, if $x$ is not one of the grid points, (5.4) is a very poor approximation to $f$, see Fig. 8. Why is this? The interpolant (5.4) only combines $\phi_n$'s with $n \geq 0$, while, according to (2.4), $f$ is, in general, a superposition of multiples of $\phi_n$'s with positive and *negative* $n$.

### 5.2.2 The good way

Having identified the reason for the failure of the interpolant (5.4), it is easy to construct a good interpolant. We certainly need a contribution involving $\phi_{-1}$. On the grid, $\phi_{-1}$ is an alias of $\phi_{M-1}$; what we can do is to replace the term $a_{M-1}\phi_{M-1}(x)$ in (5.4) (with $a$ still given by (5.5)) by $a_{M-1}\phi_{-1}(x)$. This does

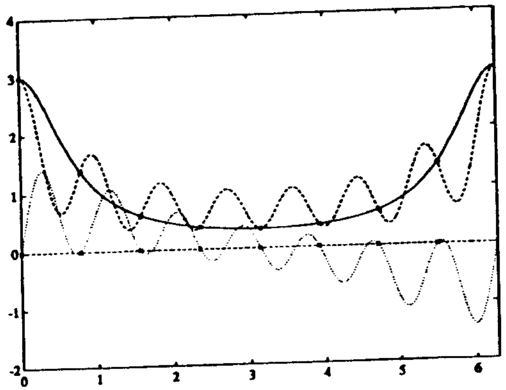

**Fig. 8.** The real function in Exercise 8 (solid line) and the interpolant (5.4)–(5.5) when $M = 8$. The interpolant is not even real-valued. The real part of the interpolant is the dashed line; the imaginary part the dotted line. At grid points, the real part of the interpolant matches the function and the imaginary part vanishes. The interpolant provides a very poor approximation indeed

not change the value of (5.4) at grid points, so that we are still interpolating $f$ on the grid. Similarly, one replaces $a_{M-1}\phi_{M-2}(x)$ by $a_{M-2}\phi_{-2}(x)$ etc.

To be precise, let us consider separately the cases where $M$ is odd and even. Assume first that $M$ is of the form $M = 2N + 1$. Then we use the interpolant given by (5.5) and

$$a_{N+1}\phi_{-N}(x) + \cdots + a_{M-1}\phi_{-1}(x) + a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_N\phi_N(x).$$

This coincides with (5.4) (and therefore with $f$) on the grid, because $\phi_{-n}$, $n = 1, \ldots, N$, is an alias of $\phi_{M-n}$.

When $M$ is even (in practice this is the commonest situation), one could consider either (Canuto et al. 1988)

$$\begin{aligned} &a_{N+1}\phi_{-N+1}(x) + \cdots + a_{M-1}\phi_{-1}(x) \\ &\quad + a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_{N-1}\phi_{N-1}(x) + a_N\phi_N(x), \end{aligned}$$

or

$$\begin{aligned} &a_N\phi_{-N}(x) + a_{N+1}\phi_{-N+1}(x) + \cdots + a_{M-1}\phi_{-1}(x) \\ &\quad + a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_{N-1}\phi_{N-1}(x), \end{aligned}$$

but I prefer to settle for the more symmetric format resulting after averaging these two expressions (Hamming 1973), i.e.,

$$\begin{aligned} &\frac{1}{2}a_N\phi_{-N}(x) + a_{N+1}\phi_{-N+1}(x) + \cdots + a_{M-1}\phi_{-1}(x) \\ &\quad + a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_{N-1}\phi_{N-1}(x) + \frac{1}{2}a_N\phi_N(x). \end{aligned}$$



We write this interpolation or *collocation* trigonometric polynomial (Fig. 9) as (a double prime in the summation means that the first and last term should be halved)

$$I_N(f) = \sum_{n=-N}^{N}{}'' \tilde{f}_n\phi_n, \quad \tilde{f}_N = \tilde{f}_{-N}, \tag{5.6}$$

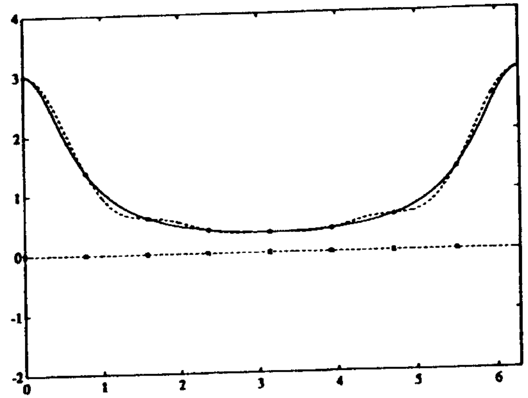

**Fig. 9.** The real function in Exercise 8 (solid line) and the interpolant (5.6) when $M = 8$. Compare with Fig. 8

with the coefficients given by

$$\tilde{f}_n = a_n, \quad n = 0, 1, \ldots, N-1, \tag{5.7a}$$
$$\tilde{f}_{-n} = a_{M-n}, \quad n = 1, 2, \ldots, N, \tag{5.7b}$$

The $a_n$ are given by (5.5).

### 5.2.3 Discrete Fourier coefficients

*Hereafter we only consider the even $M$ case, $M = 2N$.* The coefficients $\tilde{f}_n$ of the interpolant (5.6) are called the *discrete Fourier coefficients* of $f$. Note that they depend on $M$, i.e., on the specific grid under consideration; this dependence is however not reflected in the notation. We emphasize that to find the discrete Fourier coefficients of a function is an easy task. It is enough to perform the FFT in (5.5) and then rearrange via (5.7). In MATLAB, the function **fftshift(a)** rearranges the output

$$[a_0, a_1, \ldots, a_{N-1}, a_N, a_{N+1}, \ldots, a_{2N-1}]^T$$

of the function **fft** by swapping the left and right halves of the vector:

$$[a_N, a_{N+1}, \ldots, a_{2N-1}, a_0, a_1, \ldots, a_{N-1}]^T. \tag{5.8}$$

This is a handy function because it leaves the coefficients in essentially the same order as they are required in the sum in (5.6).

For vectors $\mathbf{X}(f) = [f(x_0),\dots,f(x_{M-1})]^T$ of grid values of a function $f$, it is customary to use the norm defined by

$$\|\mathbf{X}(f)\|^2 = \Delta x\,(|f(x_0)|^2 + \cdots + |f(x_{M-1})|^2)\,. \tag{5.9}$$

This differs from the usual norm $|\cdot|$ for vectors (Sect. 1.2) by the presence of the normalizing factor $\Delta x$, which is included to ensure that, for $\Delta x$ small $\|\mathbf{X}(f)\|$ is an approximation to $\|f\|$ in (2.1). This normalizing factor affects the value of the norm of the vectors, but not the value of the norm of matrices $\|A\|$, because $\|A\|$ is defined (see Sect. 1.2) in terms of quotients of vector lengths. With the definition in (5.9) the following discrete version of Parseval's identity (2.7) holds

$$\|\mathbf{X}(f)\|^2 = L \sum_{n=-N}^{N}{}'' |\tilde{f}_n|^2. \tag{5.10}$$

### 5.2.4 The trigonometric form of the collocation polynomial

It is possible to write the interpolant $I_N(f)$ in trigonometric form (cf. the derivation of (2.9)). The result is

$$I_N(f) = \tilde{c}_0(f) + \sum_{n=1}^{N-1}\left(\tilde{c}_n(f)\cos\frac{2\pi n}{L}x + \tilde{s}_n(f)\sin\frac{2\pi n}{L}x\right)$$
$$+\,\tilde{c}_N(f)\cos\frac{2\pi N}{L}x,$$

with

$$\tilde{c}_0(f) = \tilde{f}_0,$$
$$\tilde{c}_n(f) = \tilde{f}_n + \tilde{f}_{-n}, \quad n = 1,2,\dots,N,$$
$$\tilde{s}_n(f) = i(\tilde{f}_n - \tilde{f}_{-n}), \quad n = 1,2\dots,N-1.$$

The last basis function $\cos(2\pi Nx/L)$ gives rise to the vector of grid values $[1,-1,1,-1,\dots,-1]^T$ (saw-tooth behavior). Note that $\tilde{f}_N = \tilde{f}_{-N}$ results in $I_N(f)$ not including a term in $\sin((2\pi Nx)/L)$. The absentee $\sin(2\pi Nx/L)$ is, on the grid, an alias of the 0 function, so that it is just as well if it does not feature in $I_N(f)$. There are $N+1$ coefficients $\tilde{c}_n$ and only $N-1$ coefficients $\tilde{s}_n$; this makes in all $M = 2N$ coefficients, which is just right to interpolate at $M$ grid points. If $f$ is real-valued the coefficients $\tilde{c}_n(f)$ and $\tilde{s}_n(f)$ are real.

**Exercise 25** Write a program that produces the graph in Fig. 8 with arbitrary values of $M$. Run your program for $M = 4, 8, 16, \dots$ and try to make sense of the plots you get.

**Exercise 26** The saw-tooth vector $[1,-1,1,-1,\dots,-1]^T$ is one of the columns of the Fourier matrix $F_M$, $M = 2N$. Why?



**Exercise 27** Use (4.3) and (5.5)-(5.7) to prove (5.10).

## 5.3 More aliasing: discrete Fourier coefficients vs. Fourier coefficients

### 5.3.1 The relation between discrete Fourier coefficients and Fourier coefficients

There is an alternative way in which the interpolant (5.6) can be constructed. We begin with the Fourier series (2.4), that we rewrite in the form

$$f = \sum_{n=-N+1}^{N}\left(\sum_{m=-\infty}^{\infty}\hat{f}_{n+mM}\phi_{n+mM}\right), \tag{5.11}$$

i.e., we first sum the contributions involving

$$\dots,\phi_{-N+1-M},\phi_{-N+1},\phi_{-N+1+M},\dots,$$

then the contributions involving

$$\dots,\phi_{-N-M},\phi_{-N},\phi_{-N+M},\dots,$$

etc. The grid values of the right-hand side of (5.11) will not change if we replace $\phi_{n+mM}$ by its alias $\phi_n$. Therefore

$$\sum_{n=-N}^{N}{}''\left(\sum_{m=-\infty}^{\infty}\hat{f}_{n+mM}\right)\phi_n(x) \tag{5.12}$$

is, provided that the $M$ series $\sum_m \hat{f}_{n+mM}$ converge, an interpolant of $f$ on the grid. (In (5.12) we have achieved symmetry by dividing $\sum_m \hat{f}_{N+mM}$ into two halves and attaching one of the halves to $\phi_N$ and the other to its alias $\phi_{-N}$.) By uniqueness of the interpolant, (5.6) and (5.12) must coincide. This leads to the following formula (Canuto et al. 1988; Hamming 1978) relating the Fourier coefficients $\hat{f}_n$ of $f$ to the discrete Fourier coefficients $\tilde{f}_n$

$$\tilde{f}_n = \sum_{m=-\infty}^{\infty}\hat{f}_{n+mM}, \quad n = 0,\pm1,\dots,\pm N. \tag{5.13}$$

### 5.3.2 Truncation vs. interpolation

It is now expedient to compare the truncation $P_N(f)$ of the Fourier series of $f$ (see (2.6)) with the interpolant $I_N(f)$.

Both $P_N(f)$ and $I_N(f)$ are trigonometric polynomials of the form (2.8) (but $I_N(f)$ has only $2N$ degrees of freedom because $\tilde{f}_N = \tilde{f}_{-N}$). The truncation $P_N(f)$ is characterized by the property that the residual $f - P_N(f)$ is small in the sense that it is orthogonal to the $2N+1$ functions $\phi_n$, $n = 0,\pm1,\dots,\pm N$. The interpolant $I_N(f)$ is characterized by the property that the residual $f - I_N(f)$ is small in the sense that it vanishes at $2N$ grid points.

The truncation $P_N(f)$ has coefficients $\hat{f}_{-N}, \ldots, \hat{f}_N$ that are independent of the coefficients $\hat{f}_n$ with $|n| > N$, i.e., independent of the modes in $f$ with $|n| > N$. On the other hand, the Fourier coefficients $\tilde{f}_n$ with $|n| > N$ do contribute to the interpolant via (5.13). The mode $\hat{f}_n \phi_n(x)$ of the Fourier series 'feeds' the alias mode involving the basis function $\phi_{n'}$ for which $n \equiv n'$.

We noted already that it is easy to find the coefficients $\tilde{f}_n$. On the other hand, the coefficients $\hat{f}_n$ are defined by the integrals (2.5), which in practice should be evaluated numerically (see Exercise 28).

As we discussed in Sect. 2.3, the smoothness of $f$ governs the decay of the $\hat{f}_n$ and hence the velocity of the convergence of $P_N(f)$ to $f$. The formulae (5.13) may be used (Tadmor 1986) to show that in a like manner, smoother functions have discrete Fourier coefficients that decay faster. Also, the smoother $f$ is the faster the convergence of $I_N(f)$ to $f$.

### 5.3.3 The sampling theorem

For functions $f$ that satisfy $\hat{f}_n = 0$ for $|n| \geq N$, it is true that $I_N(f) = f = P_N(f)$. Such functions, being equal to $I_N(f)$ can be reconstructed from its stroboscopic or grid samples $\mathbf{X}(f)$ through (5.5)-(5.7). When written in trigonometric form, such functions only possess harmonics with frequencies $n/L$ below the upper bound $N/L$. Now, since $\Delta x = L/(2N)$, the frequency upper bound $N/L$ equals $1/(2\Delta x)$; this is called (Hamming 1973) the *Nyquist frequency*. Correspondingly, the periods of the harmonics of $f$ have a *lower* bound $2\Delta x$. The spacing $\Delta x$ between two consecutive samples must be smaller than half of the smallest period of the harmonics of $f$, if $f$ is to be reconstructed from its grid values. The function $\sin(2\pi N x/L)$ whose period is exactly $2\Delta x$ cannot be distinguished on the grid from the 0 function.

**Exercise 28** Show that $\hat{f}_n$, $n = 0, \pm 1, \ldots, \pm N$, is a linear combination of grid values of $f$ and that this linear combination can be seen as a numerical approximation to the integral (2.5) defining $\hat{f}_n$.

**Exercise 29** Use (5.13) to compute, for arbitrary $M = 2N$ the discrete Fourier coefficients of the function in Exercise 8. Show that these coefficients decay exponentially as a function of $|n|$, at a rate which does not depend on $M$.

**Exercise 30** Set $M = 4$ and compute the discrete Fourier coefficients of the function in Exercise 8 via (5.5), (5.7). Do you get the same results you found in Exercise 29?



# 6. Fourier Analysis of Finite-Difference Algorithms: the Time-Continuous Case

## 6.1 Spatial discretizations of initial value problems

### 6.1.1 The discretization

We now study the numerical solution of the periodic problem (3.1)-(3.2). For the sake of clarity, it is not advisable to look at the 'general' equation (3.1) and we base our presentation on a model: the advection equation with velocity $c = 1$ (see Sect. 3.3)

$$\partial_t u = -\partial_x u. \tag{6.1}$$

In this section we look at semidiscrete (discrete $x$, continuous $t$) approximations to (6.1). In a finite-difference approach, the variable $x$ is discretized as in Section 5.1 and we look for approximations $U_n(t)$ to the solution values $u(x_n, t)$, $n = 0, \pm 1, \pm 2, \ldots$, $t \geq 0$. By periodicity $U_n(t) = U_m(t)$ whenever $n \equiv m$ and there are really $M$ unknowns $U_n(t)$. These are collected into a vector $\mathbf{U}(t)$. The operator $\partial_x$ is replaced by a suitable finite difference formula, for instance (Mitchell and Griffiths 1980)

$$\partial_x u(x_n, t) \approx \frac{u(x_{n+1}, t) - u(x_n, t)}{\Delta x},$$

(forward differences), or alternatively

$$\partial_x u(x_n, t) \approx \frac{u(x_n, t) - u(x_{n-1}, t)}{\Delta x},$$

(backward differences), or

$$\partial_x u(x_n, t) \approx \frac{u(x_{n+1}, t) - u(x_{n-1}, t)}{2\Delta x}$$

(central differences).

The numerical approximations are then asked to satisfy

$$\frac{d}{dt} U_n(t) = -\frac{U_{n+1}(t) - U_n(t)}{\Delta x}, \quad n = 0, 1, \ldots, M-1,$$

or

$$\frac{d}{dt} U_n(t) = -\frac{U_n(t) - U_{n-1}(t)}{\Delta x}, \quad n = 0, 1, \ldots, M-1,$$

or

$$\frac{d}{dt} U_n(t) = -\frac{U_{n+1}(t) - U_{n-1}(t)}{2\Delta x}, \quad n = 0, 1, \ldots, M-1.$$

Thus $\mathbf{U}(t)$ is asked to satisfy the system of differential equations

$$\frac{d}{dt} \mathbf{U}(t) = A \mathbf{U}(t), \quad t \geq 0, \tag{6.2}$$

where $A$ takes one of the forms

$$A = \begin{bmatrix} 1/\Delta x & -1/\Delta x & 0 & \cdots & 0 \\ 0 & 1/\Delta x & -1/\Delta x & \cdots & 0 \\ 0 & 0 & 1/\Delta x & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1/\Delta x & 0 & 0 & \cdots & 1/\Delta x \end{bmatrix}, \tag{6.3a}$$

$$A = \begin{bmatrix} -1/\Delta x & 0 & 0 & \cdots & 1/\Delta x \\ 1/\Delta x & -1/\Delta x & 0 & \cdots & 0 \\ 0 & 1/\Delta x & -1/\Delta x & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1/\Delta x \end{bmatrix}, \tag{6.3b}$$

$$A = \begin{bmatrix} 0 & -1/(2\Delta x) & 0 & \cdots & 1/(2\Delta x) \\ 1/(2\Delta x) & 0 & -1/(2\Delta x) & \cdots & 0 \\ 0 & 1/(2\Delta x) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1/(2\Delta x) & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{6.3c}$$

More sophisticated choices of $A$ are of course possible (Exercise 31).

The matrices in (6.3) and other matrices arising from discretization of (6.1) or more generally other partial differential equations of the form (3.1) have some important features. They are *sparse:* even though they contain $O(M^2)$ entries, the number of nonzero entries is only $O(M)$. This makes a matrix vector product much cheaper than in the general case. Furthermore the matrices are *circulant* (Sect. 4.2). This is a consequence of the periodicity of the boundary conditions and (3.1) having constant coefficients.

The system (6.2) needs an initial condition $\mathbf{U}(0)$, which is usually taken to be the vector $\mathbf{X}(u^0)$ obtained by restriction of the initial condition $u^0$ in (3.2), i.e.,

$$\mathbf{U}(0) = \mathbf{X}(u^0). \tag{6.4}$$

In practice one may integrate (6.2), (6.4) with a numerical solver for ordinary differential equations.

The discretization of the spatial variables may alternatively be carried out by finite elements (Strang and Fix 1973). While in cases with several space variables finite elements are more versatile in dealing with the geometry of the problem, in one space dimension there is not much difference between finite elements and finite differences. Finite elements would also lead to a system of the form (6.2); in fact one may think (Mitchell and Griffiths 1980) of the finite element method as of a method to generate good finite difference schemes. In what follows we only deal with finite differences, in the understanding that most of the material can be extended to finite elements.



### 6.1.2 Consistency

Our aim is now to estimate the error perpetrated by taking $\mathbf{U}(t)$ as an approximation to $u(x,t)$. More precisely we want to measure the size of the vector $\mathbf{E}(t) = \mathbf{U}(t) - \mathbf{X}(u(\cdot,t))$ called the *global error*. This size is defined by (see (5.9))

$$\|\mathbf{E}\| = \sqrt{\sum_{n=0}^{M-1} \Delta x |U_n(t) - u(x_n,t)|^2}. \tag{6.5}$$

In order to bound (6.5) an indirect approach is followed, which uses the ideas of *consistency* and *stability* (Richtmyer and Morton 1976; Sanz-Serna 1991).

Consistency refers to the relation between the partial differential equation *problem* being solved and the numerical *problem* (6.2). This is not to be confused with the question we want solved, namely with the relation between the *solutions* of the partial differential equation problem and (6.2).

To study consistency we substitute in (6.2) the vector $\mathbf{X}(u(\cdot,t))$ of the grid values of the true solution $u$; this gives rise to a residual

$$\mathbf{L}(t) = \frac{d}{dt}\mathbf{X}(u(\cdot,t)) - A\mathbf{X}(u(\cdot,t)) \tag{6.6}$$

called the *truncation* or *local* error.

With the choice (6.3a), the $n$-th component of $\mathbf{L}(t)$ is given by

$$L_n(t) = \partial_t u(x_n,t) + \frac{u(x_{n+1},t) - u(x_n,t)}{\Delta x}, \tag{6.7}$$

or, after Taylor expanding,

$$L_n(t) = \partial_t u(x_n,t) + \frac{1}{\Delta x}\left[\Delta x \partial_x u(x_n,t) + \frac{\Delta x^2}{2}\partial_{xx} u(x_n,t) + \cdots\right].$$

Since $u$ satisfies (6.1),

$$L_n(t) = \frac{\Delta x}{2}\partial_{xx} u(x_n,t) + \cdots = O(\Delta x), \quad \Delta x \to 0.$$

The definition of $\|\cdot\|$ (see Exercise 27 and (6.5)) involves summation of $M = L/\Delta x$ terms, but also includes a $\Delta x$ factor. Therefore $O(\Delta x)$ entries imply $O(\Delta x)$ norms and

$$\|\mathbf{L}(t)\| = O(\Delta x), \quad \Delta x \to 0. \tag{6.8}$$

The conclusion is that, with the choice (6.3a), the problem (6.2) is an approximation to the true problem (6.1), in the sense that solutions $u$ of (6.1) satisfy (6.2) except for an $O(\Delta x)$ remainder. The finite-difference approximation (6.2) is then said to be consistent. Since only the first power of $\Delta x$ appears in (6.8), we say that (6.2)–(6.3a) is consistent *of the first order*.

A similar argument reveals that (6.2) with (6.3b) is also consistent of the first order. For (6.3c),

$$\|\mathbf{L}(t)\| = O(\Delta x^2), \quad \Delta x \to 0,$$

this is consistency of the second order.

### 6.1.3 Stability

We say that (6.2) is *stable*, if, for each finite interval $0 \le t \le T$, a constant $C(T)$ can be found such that, for all solutions $\mathbf{U}(t)$ of (6.2),

$$\|\mathbf{U}(t)\| \le C(T)\|\mathbf{U}(0)\|, \quad 0 \le t \le T; \tag{6.9}$$

small initial conditions lead to small solutions. What is important here is that $C(T)$ has to be independent, not only of the initial condition $\mathbf{U}(0)$, but also of the parameter $\Delta x$. Independence of $\Delta x$ is a delicate business: as $\Delta x \to 0$, the entries of the matrix $A$ in (6.2) blow up and one would expect that $\mathbf{U}(t)$ may also grow.

There are two things I would like to point out:

(i) The requirement (6.9) is a discrete analogue of the well-posedness estimate (3.8). In fact (3.8) implies

$$\|u(\cdot, t)\| \le C(T)\|u(\cdot, 0)\|, \quad 0 \le t \le T,$$

with $C(T) = \exp(\alpha(P(\partial_x))T)$.

(ii) Since $\mathbf{U}(t) = \exp(tA)\mathbf{U}(0)$, (6.9) is equivalent to

$$\|e^{tA}\| \le C(T), \quad 0 \le t \le T. \tag{6.10}$$

For the choice (6.3c), (6.2) is stable. In fact, since $A$ is skew-symmetric, $\exp(tA)$ is unitary,

$$(e^{tA})^* = e^{tA^*} = e^{-tA} = (e^{tA})^{-1},$$

(Sect. 1.2) so that (6.10) holds with $C(T) = 1$. The stability for the choices (6.3a), (6.3b) is discussed later.

### 6.1.4 Convergence

Subtraction of (6.6) from (6.2) leads to the following system of differential equations for the global error we wish to estimate

$$\frac{d}{dt}\mathbf{E}(t) = A\mathbf{E}(t) - \mathbf{L}(t). \tag{6.11}$$

This is similar to the system (6.2) satisfied by the numerical solution itself; the difference is that (6.2) is homogeneous while (6.11) has the truncation error $\mathbf{L}(t)$ as a forcing term.

The solution of (6.11) may be written via the variation of constants formula (Strang 1986)

$$\mathbf{E}(t) = e^{tA}\mathbf{E}(0) - \int_0^t e^{(t-s)A}\mathbf{L}(s)\, ds. \tag{6.12}$$

Thus $\mathbf{E}(t)$ is the sum of two contributions. The first $\exp(tA)\mathbf{E}(0)$ represents the time evolution of the initial global error $\mathbf{E}(0)$; if $\mathbf{U}(0)$ is taken as in (6.4), this first contribution vanishes. The second contribution is a superposition of terms $-\exp((t-s)A)\mathbf{L}(s)$; each of the terms being superposed is the effect at time $t$ of the forcing term $-\mathbf{L}(s)$ acting as an initial condition at time $s$ (Duhamel's principle).



Assume that the numerical method (6.2) is stable and consistent of order $p$, then, using (6.10) in (6.12),

$$\begin{aligned}\|\mathbf{E}\| &\le \|e^{tA}\|\|\mathbf{E}(0)\| + \int_0^t \|e^{(t-s)A}\|\|\mathbf{L}(s)\|\, ds \\ &\le C(T)\|\mathbf{E}(0)\| + TC(T)O(\Delta x^p), \quad 0 \le t \le T,\end{aligned}$$

and, with the standard choice (6.4) for $\mathbf{U}(0)$,

$$\|\mathbf{E}\| \le TC(T)O(\Delta x^p) = O(\Delta x^p), \quad 0 \le t \le T;$$

the global errors decay as $O(\Delta x^p)$ as the grid is refined. This is called *convergence of order p*. To sum up, we have just proved that ***stability and consistency of order p yield convergence of order p***. Conversely, both stability and consistency are necessary for convergence; this is what the celebrated Lax equivalence theorem says (Richtmyer and Morton 1967; Sanz-Serna and Palencia 1985).

There is a subtle point we should not avoid (Richtmyer and Morton 1967; Sanz-Serna 1985). The investigation of consistency (see Sect. 6.1.2) is based on Taylor expansions which may only be carried out if the true solution $u$ is smooth enough. (More precisely if $u$ has continuous derivatives up to a given order depending on the specific finite-difference scheme being investigated; for instance the derivation of (6.8) required the existence and continuity of the functions $\partial_t u$, $\partial_{xx} u$.) Hence we have really proved convergence of order $p$ only for smooth solutions. What is the situation for nonsmooth solutions? With the square-wave function in Exercise 3 as an initial condition and the forward difference matrix $A$ in (6.3a), the truncation error $L_n(t)$ in (6.7) equals $-2/\Delta x$ when $x_n = \pi$ and $t = 0$; the finer the grid the worse the approximation! Nevertheless it can be shown that, for a scheme that is stable and consistent of order $p$, convergence holds for *all* solutions, regardless of their smoothness. Smooth solutions have $\|\mathbf{E}(t)\| = O(\Delta x^p)$, other solutions have $\|\mathbf{E}(t)\| = O(\Delta x^q)$ with $q < p$; the exact value of $q$ depends on the exact smoothness of $u$. In the square-wave example convergence is $O(\Delta x^{1/4})$ (Sanz-Serna, 1985).

**Exercise 31** For the problem (6.1) write a finite-difference formula of the form

$$\partial_x u(x_n, t) \approx \alpha_2 u(x_{n+2}, t) + \alpha_1 u(x_{n+1}, t) + \alpha_0 u(x_n, t) + \alpha_{-1} u(x_{n-1}, t) + \alpha_{-2} u(x_{n-2}, t),$$

with the $\alpha_m$ chosen to achieve the highest possible order of consistency. Analyze the stability and convergence of the resulting scheme.

**Exercise 32** For the heat equation construct difference schemes of the form (6.2) with orders of consistency 2, 4 or 6.

## 6.2 The Von Neumann stability analysis

We have not yet discussed the stability of (6.2) when $A$ is given by (6.3a) or (6.3b). This stability will now be investigated by Fourier analysis.
With the choice (6.3a) or (6.3b), the matrix $A$ in (6.2) is, as any other circulant matrix (see Sect. 4.2), normal (Section 1.2). Therefore $\|\exp(tA)\| = \exp(t\alpha(A))$, with $\alpha(A)$ the spectral abscissa of $A$ (Section 1.2). Then the stability condition (6.10) holds if and only if

$$\sup_{\Delta x} \alpha(A) < \infty. \qquad (6.13)$$

This is called the Von Neumann condition and is a direct analogue of the well posedness condition $\alpha < \infty$ in Sect. 3.2 but here there is an extra parameter $\Delta x$.

The eigenvalues of the most general circulant matrix were found in (4.6) by Fourier analysis. For (6.3b) (backward differences) the eigenvalues are

$$\sigma_n = -\frac{1}{\Delta x} + \frac{1}{\Delta x}\exp\left(\frac{2\pi(M-1)ni}{M}\right), \quad n = 0, 1, \ldots, M-1;$$

or, by periodicity,

$$\sigma_n = -\frac{1}{\Delta x} + \frac{1}{\Delta x}\exp\left(-\frac{2\pi ni}{M}\right), \quad n = 0, 1, \ldots, M-1; \qquad (6.14)$$

this leads to $\alpha(A) \leq 0$ and therefore to stability with stability constant $C(T) = 1$.

On the other hand, for (6.3a) (forward differences) the eigenvalues are

$$\sigma_n = \frac{1}{\Delta x} - \frac{1}{\Delta x}\exp\left(\frac{2\pi ni}{M}\right), \quad n = 0, 1, \ldots, M-1; \qquad (6.15)$$

thus $2/\Delta x$ is an eigenvalue ($n = N = M/2$) and $\alpha(A) \geq 2/\Delta x$. As the grid is refined for fixed $t$, $\|\exp(tA)\| \geq \exp(2/\Delta x)$ grows exponentially and we have instability. By the Lax equivalence theorem the scheme (6.2)–(6.3a) is not convergent, in spite of the fact that it is a 'reasonable' discretization of (6.1). I would like to emphasize that it is the growth of $\exp(tA)$ for fixed $t$ as $\Delta x \to 0$ which prevents stability and convergence. This growth is not to be confused with growth for fixed $\Delta x$ and $t \to \infty$, see Exercise 34.

Let us summarize. We use Fourier analysis to find the eigenvalues $\sigma_n$ of the system (6.2) we are investigating. The eigenvalues of $\exp(tA)$ are then $\exp(t\sigma_n)$ and, *due to the normality of* $A$, the norm of $\exp(tA)$, which we want bounded, coincides with the eigenvalue $\exp(t\sigma_n)$ with maximum modulus. This arises from the $\sigma$ with maximum real part.

**Exercise 33** Use the Von Neumann method to investigate the stability of the difference schemes constructed in Exercises 31 and 32.

**Exercise 34** Consider the equation $\partial_t u = \partial_x u + u$. Study the well-posedness of the corresponding periodic initial value problem (cf. Exercise 16). Discretize this problem by central differences and analyze the stability and convergence of the resulting discretization. Note that $\|\exp(tA)\|$ grows with $t$ and, nevertheless, the numerical method is stable.



**Exercise 35** Discretize the system (3.7) by the finite difference scheme

$$\frac{d}{dt}V_n(t) = c\frac{W_n(t) - W_{n-1}(t)}{\Delta x}, \quad n = 0, 1, \ldots, M-1,$$
$$\frac{d}{dt}W_n(t) = c\frac{V_{n+1}(t) - V_n(t)}{\Delta x}, \quad n = 0, 1, \ldots, M-1.$$

Introducing the vector of unknowns $\mathbf{U} = [V_0, W_0, \ldots, V_{M-1}, W_{M-1}]^T$, write the difference equations in the format (6.2), with $A$ of dimension $2M$ ($M$ is, as always, the number of grid points). Is the matrix $A$ you obtain a circulant matrix? The answer should be no. Use Fourier analysis to find the eigenvalues of the matrix $A$ you have found. The result should be $\pm(2ci/\Delta x)\sin(\pi n\Delta x/L)$, $n = 0, 1, \ldots, M-1$ (Hint: Assume that the eigenvectors have $V_n = A\exp((2\pi ni\Delta x)/L)$ and $W_n = B\exp((2\pi ni\Delta x)/L)$ and substitute in $A\mathbf{U} = \sigma\mathbf{U}$.) In general, for systems of partial differential equations the matrix is *not* circulant, but is such that its eigenvalues can be explicitly found by Fourier analysis. The condition that the real part of the eigenvalues of $A$ should be bounded above is still necessary for stability, because for any matrix $\|\exp(tA)\| \geq \rho(\exp(tA)) = \exp(t\alpha(A))$, $t > 0$. However the study of the eigenvalues of $A$ is not in general sufficient for stability, because if $A$ is not normal the norm $\|\exp(tA)\|$ may be strictly larger than the spectral radius $\rho(\exp(tA)) = \exp(t\alpha(A))$. For this reason for linear, constant coefficient *systems* of partial differential equations with periodic boundary conditions, the Von Neumann condition (6.13) is necessary but not sufficient for stability (Richtmyer and Morton 1967).

## 6.3 The roles of stability and consistency from a Fourier viewpoint

The Von Neumann stability test provides the easiest application of Fourier methods to the numerical analysis of initial value problems. There are other applications certainly worth studying. For instance, it is useful to gain insight, via Fourier analysis, into the equivalence between convergence and consistency plus stability.

We still look at the advection equation (6.1) solved by any of the methods in (6.2)–(6.3). The theoretical solution was found in (3.6) and is given by

$$u(x,t) = \sum_{n=-\infty}^{\infty} \exp(\mu_n t)\hat{u}_n^0 \phi_n(x), \quad \mu_n = -2\pi ni/L.$$

We evaluate $u$ at grid points in order to make it possible a comparison with the numerical solution $\mathbf{U}(t)$. The result is

$$\mathbf{X}(u(\cdot,t)) = \sum_{n=-\infty}^{\infty} \exp(\mu_n t)\hat{u}_n^0 \mathbf{X}(\phi_n), \quad \mu_n = -2\pi ni/L. \tag{6.16}$$

We could now use the aliasing relations (5.2) to replace each $\mathbf{X}(\phi_n)$ by an $\mathbf{X}(\phi_m)$, with $-N \le m \le N$; we performed such a replacement when we wrote formula (5.11). However we prefer to leave (6.16) as it stands.

We next write the numerical solution $\mathbf{U}(t)$ in a format similar to (6.16). Before we do this, let me go back to the system (4.4) with *general* circulant matrix. One easily checks that, for $n = 0, \pm 1, \pm 2, \ldots$, the vector $\mathbf{X}(\phi_n)$ in (5.1) is an eigenvector of the matrix in (4.4); the corresponding eigenvalue is

$$\sigma_n = a_0 + a_1 \exp\left(\frac{2\pi ni}{M}\right) + \cdots + a_{M-1}\exp\left(\frac{2\pi(M-1)ni}{M}\right). \tag{6.17}$$

Does this mean that the $M \times M$ matrix $A$ possesses infinitely many eigenvectors/values? Certainly it does not. If $n \equiv m$, then $\mathbf{X}(\phi_n) = \mathbf{X}(\phi_m)$, so that we have only found $M$ distinct eigenvectors. Indeed, upon recalling (5.3), we see that the $\mathbf{X}(\phi_n)$ are scaled versions of the columns $\mathbf{V}_n$ of $F_M^{-1}$ we resorted to in Sect. 4.2. Correspondingly, in (6.17) $\sigma_n = \sigma_m$ whenever $n \equiv m$, because $\exp(2\pi\ell ni/M)$ is an $M$-periodic function of $n$. Comparison with (4.6) gives us the reassuring conclusion that we have again found the $M$ eigenvectors/functions we had in Sect. 3.2. The difference is that now we let $n$ to be arbitrary in (6.17), while in (4.6) $n$ was between 0 and $M-1$. The solution of (4.4) with initial condition $\mathbf{X}(u^0)$ is then

$$\sum_{n=-\infty}^{\infty} \exp(\sigma_n t)\hat{u}_n^0 \mathbf{X}(\phi_n). \tag{6.18}$$

This certainly has the correct value at time $t = 0$; furthermore, it satisfies (4.4) as one easily checks by substitution in the differential system. We could use of the aliasing relations (5.2) to rewrite (6.18) as a sum with only $M$ terms; the result would be the solution (4.11) we found in Sect. 4.2.

It is time to leave the general problem (4.4) and return to our finite-difference method (6.2)–(6.3). The numerical solution with initial condition (6.4) is given by (6.18) with the $\sigma_n$ equal to the eigenvalues of the matrix $A$ corresponding to the specific choice of finite-difference method; for instance for the central-difference method (6.3c), the eigenvalues (6.17) are readily found to be

$$\sigma_n = -\frac{i}{\Delta x}\sin\left(\frac{2\pi n}{M}\right) = -\frac{i}{\Delta x}\sin\left(\frac{2\pi n\Delta x}{L}\right). \tag{6.19}$$

For backward and forward differences the eigenvalues were found in (6.14) and (6.15).

Subtraction of (6.16) from (6.18) provides the expression for the global error

$$\mathbf{E}(t) = \sum_{n=-\infty}^{\infty} \left[\exp(\sigma_n t) - \exp(\mu_n t)\right] \hat{u}_n^0 \mathbf{X}(\phi_n). \tag{6.20}$$



This is the representation we wish to discuss. Let us first fix a value of $n$ and look at the corresponding term in the series (6.20). We have an exponent $\mu_n$ coming from the theoretical solution and an exponent $\sigma_n$ from the numerical solution. To be specific, assume that we use central differences with $\sigma_n$ given by (6.19). As the grid is refined ($\Delta x \to 0$), Taylor expansion in (6.19) yields

$$\begin{aligned}\sigma_n &= -\frac{i}{\Delta x}\left(\frac{2\pi n\Delta x}{L}\right) + \frac{1}{6}\frac{i}{\Delta x}\left(\frac{2\pi n\Delta x}{L}\right)^3 - \cdots \\ &= \mu_n + \frac{4\pi^3 n^3 i}{3L^3}\Delta x^2 + \cdots,\end{aligned}$$

so that $\sigma_n$ approaches $\mu_n$. A fixed mode becomes better and better approximated as $\Delta x \to 0$. This is the Fourier analysis expression of the consistency of the method. In fact, above, $\sigma_n = \mu_n + O(\Delta x^2)$ because (6.3c) leads to consistency of the second order. For forward or backward differences $\sigma_n = \mu_n + O(\Delta x)$. This is the good news: consistency guarantees that all is well as $\Delta x \to 0$ with $n$ fixed.

The bad news is that, on any fixed grid you may be using, there are numbers $n$ for which $\sigma_n$ and $\mu_n$ are grossly different. This is made clear in Fig. 10, that should be studied carefully. While the figure refers to (6.2), (6.3c), a similar discussion holds for the choices (6.3a) or (6.3b).

If, for each $\Delta x$, there are terms in the series (6.20) for which $\exp(\sigma_n t) - \exp(\mu_n t)$ is large, how is it possible to get convergence, i.e. small $\mathbf{E}(t)$? It is the $\hat{u}_n^0$ that come to the rescue: they must decay as $|n| \to \infty$, because the series in (2.7) converges under the only assumption that $u^0$ is in $\mathcal{L}^2[0, L]$. Furthermore, we noticed in Sect. 2.3, that, the smoother the initial datum $u^0$, the faster the $\hat{u}_n^0$ decay. It is this decay that is implicit at the heart of convergence and explains why the rate of convergence decreases for nonsmooth solutions. The mathematical details are as follows (Richtmyer and Morton 1967). Assume, that, at a given time $t > 0$, we want to make the norm of $\mathbf{E}(t)$ less than a given small quantity via a suitable choice of $\Delta x$. One begins by finding an index $\nu$ for which

$$\sum_{|n|>\nu} [\hat{u}_n^0]^2 \tag{6.21}$$

is small. This is possible because of the convergence of the series in (2.7). Once this $\nu$ is known, we take $\Delta x$ small enough to ensure that

$$\sum_{|n|\le\nu} \left[\exp(\sigma_n t) - \exp(\mu_n t)\right] \hat{u}_n^0 \mathbf{X}(\phi_n)$$

is small; this is possible for a consistent scheme because the sum involves a finite number of modes, and for each mode $\sigma_n$ approaches $\mu_n$ as the grid is refined. This leaves us with the remaining terms

$$\sum_{|n|>\nu} \left[e^{\sigma_n t} - e^{\mu_n t}\right] \hat{u}_n^0 \mathbf{X}(\phi_n). \tag{6.22}$$

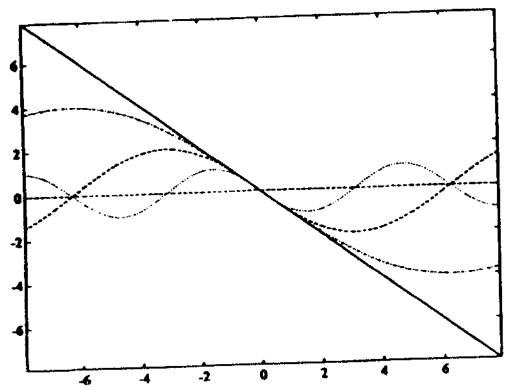


Fig. 10. The horizontal axis corresponds to the wave number $2\pi n/L$. The solid line gives the imaginary part of the exponent $\mu_n$ in the theoretical solution (6.16). The sinusoids give the imaginary part of of the exponent $\sigma_n$ in the central difference numerical solution (6.18)–(6.19). The dotted line corresponds to $\Delta x = 1$, the dashed line has $\Delta x = 1/2$ and the dash-dotted line has $\Delta x = 1/4$. Let us first look at a fixed location in the horizontal axis (i.e., a fixed wave number): the numerical $\sigma_n$ approach, as $\Delta x \to 0$, the theoretical $\mu_n$. This is a reflection of the consistency of the scheme. Let us then look, for fixed $\Delta x$, at $\sigma_n$ as a function of the wave number. The numerical $\sigma_n$ is close to $\mu_n$ only for those wave numbers $2\pi|n|/L$ that are small relatively to $(\Delta x)^{-1}$. In other words, only wave lengths that are large relatively to the grid spacing $\Delta x$ are well approximated on any given grid. For instance, when the wave length $L/n$ is $12\Delta x$, one has $\sigma_n = -i/(2\Delta x)$ and $\mu_n = -i\pi/(6\Delta x)$; since $\pi/6 = 0.52$, this means a relative error of less than 5%. When the wave length is only $4\Delta x$, $\mu_n = -i\pi/(2\Delta x)$ and $\sigma_n = -i\Delta x$ with relative error of about 50%. For the Nyquist wave number given by $\pi/\Delta x$ (saw tooth mode) the approximation is completely wrong because $\sigma_n = 0$. Note (see the dotted line) that $\sigma_n$ as a function of the wave number is a periodic function that repeats itself after the Nyquist value. This periodicity is a consequence of aliasing. Smaller $\Delta x$ lead to larger Nyquist wave numbers; the grid supports more essentially different modes

Here the $\hat{u}_n^0$ are small thanks to (6.21), and the term $\exp(\mu_n t)$ is bounded by the well-posedness of the theoretical problem. Therefore (6.22) will be small, provided that $\exp(\sigma_n t)$ is under control. Now the control of the size of $\exp(\sigma_n t)$ coincides, as we discussed in the previous section, with the issue of the stability of the method.

Let me summarize. Consistent finite difference schemes approximate badly the small wave length modes. Any reasonable initial condition is relatively poor in small wave length modes and the numerical method will perform satisfactorily if the small wave lengths, that are being misrepresented, are controlled and remain small. This control corresponds to stability.



**Exercise 36** Draw a figure similar to Fig. 10 for the method constructed in Exercise 31. Compare with Fig. 10. Observe the advantages of the high-order method (Kreiss and Oliger 1973; Fornberg 1975, 1987, 1990).

## 6.4 Numerical dissipation and numerical dispersion

Let us again consider the central difference solution (6.18)–(6.19) that we write in trigonometric form as (see Sect. 3.3)

$$A_0\mathbf{X}(1) + \sum_{n=1}^{\infty} A_n \mathbf{X}\left(\cos\left[\frac{2\pi n}{L}x - \frac{1}{\Delta x}\sin\left(\frac{2\pi n \Delta x}{L}\right)t - \psi_n\right]\right).$$

With the terminology of Sect. 3.3, the numerical solution to the dispersionless problem (6.1) thus turns out to be a dispersive wave with dispersion relation

$$\omega_n = \frac{1}{\Delta x}\sin(\kappa_n \Delta x), \quad \kappa_n = \frac{2\pi n}{L};$$

in the limit $\Delta x \to 0$ one has $\omega_n = \kappa_n$, which is the correct relation for the partial differential equation (6.1). For any (nonzero) $\Delta x$, no matter how small, there is a dependence on $\kappa_n$ of the phase speed $\omega_n/\kappa_n$. For instance, small wave numbers $\kappa_n$ travel with velocities close to the correct value 1, while the wave number $\kappa_n = \pi/\Delta x$ (the Nyquist value corresponding to the sawtooth mode) stands still at 0 phase velocity. As a result of the spurious dispersion introduced by the process of discretization, the shape of the numerical solution will change with $t$ (cf. Fig. 5).

Spurious, numerically induced dispersion is not the only problem discretization may bring. Spurious numerical dissipation is also a common occurrence, see Exercise 37. The study of the properties of dissipation and dispersion of numerical schemes is a key ingredient in the analysis of the schemes and is related to several theories from physics (Vichnevetsky 1987a, 1987b, 1989, 1990, 1992; Trefethen 1982, 1983).

**Exercise 37** Write in trigonometric form the solution of the backward difference method (6.2), (6.3b) and notice the spurious dissipative behavior.

# 7. Fourier Analysis of Finite-Difference Algorithms: the Fully-Discrete Case

## 7.1 Full discretization of initial value problems

### 7.1.1 Time discretization

In Sect. 6 the variable $t$ remained continuous. We now study fully-discrete numerical methods where $t$ is also discretized (the term 'fully' indicates that all independent variables become discrete). We denote by $\Delta t$ the time increment and consider the grid values $t^m = m\Delta t$, $m = 0, 1, 2, \dots$

In order not to blur the exposition, I shall still concentrate on the model equation (6.1); however the material has wider applicability. A fully-discrete finite-difference scheme for (6.1) may easily be obtained by replacing $\partial_t u$ and $\partial_x u$ by suitable increment quotients. It is however more advisable not to discretize $t$ and $x$ simultaneously. We proceed in two stages. First we discretize $x$ and obtain, say, one of the methods in (6.2)–(6.3). Once the system of differential equations (6.2) is available, we discretize its independent variable $t$. The discretization of $t$ then takes place in the context of a system of ordinary differential system rather than of a partial differential equation (Sanz-Serna and Verwer 1989). Any of the standard numerical methods (Lambert 1991; Hairer and Wanner 1991, Hairer et al. 1993, Sanz-Serna and Calvo 1994) for systems of the form

$$\frac{d}{dt}\mathbf{Y}(t) = \mathbf{F}(t, \mathbf{Y}(t)), \qquad (7.1)$$

is in principle eligible. Some well-known one-step possibilities are the *Euler rule*

$$\mathbf{Y}^{m+1} = \mathbf{Y}^m + \Delta t \mathbf{F}(t^m, \mathbf{Y}^m), \quad m = 0, 1, 2, \dots, \qquad (7.2)$$

($\mathbf{Y}^m$ is the numerical approximation to $\mathbf{Y}(t^m)$), the *backward Euler rule*

$$\mathbf{Y}^{m+1} = \mathbf{Y}^m + \Delta t \mathbf{F}(t^{m+1}, \mathbf{Y}^{m+1}), \quad m = 0, 1, 2, \dots, \qquad (7.3)$$

and the *trapezoidal rule*

$$\mathbf{Y}^{m+1} = \mathbf{Y}^m + \frac{\Delta t}{2}\mathbf{F}(t^{m+1}, \mathbf{Y}^{m+1}) + \frac{\Delta t}{2}\mathbf{F}(t^m, \mathbf{Y}^m), \quad m = 0, 1, 2, \dots. \qquad (7.4)$$

Higher-order methods include the celebrated (but obsolete) classical Runge-Kutta fourth-order method.

The Euler method is *explicit;* it provides a formula for finding $\mathbf{Y}^{m+1}$ once the preceding $\mathbf{Y}^m$ is available. The backward Euler and trapezoidal rules are *implicit:* to find each $\mathbf{Y}^{m+1}$ one has to solve a system of algebraic equations, which may be a costly task. ***Multistep methods*** where $\mathbf{Y}^{m+1}$ is linked to $\mathbf{Y}^m$, $\mathbf{Y}^{m-1}$, etc. are possible and may be very efficient; I am sorry they cannot be considered here.

All reasonable one-step methods (including (7.2)–(7.4)) have the property that, when the system (7.1) takes the simple linear form



$$\frac{d}{dt}\mathbf{Y}(t) = B\mathbf{Y}(t), \qquad (7.5)$$

$B$ a constant matrix, the approximation $\mathbf{Y}^{m+1}$ may be obtained from $\mathbf{Y}^m$ by matrix multiplication. More precisely

$$\mathbf{Y}^{m+1} = R(\Delta t B)\mathbf{Y}^m, \quad m = 0, 1, 2, \dots, \qquad (7.6)$$

where $R(z)$ is a rational function (see Sect. 1.2) that depends on the method but not on the specific problem of the form (7.5). For instance, $R(z) = 1 + z$ for the Euler rule, while the backward Euler rule has $R(z) = 1/(1 - z)$ and for the trapezoidal rule $R(z) = (1 + z/2)/(1 - z/2)$. Thus with each one-step method one associates a rational function; it turns out that many theoretical properties of the method may be studied by looking at the corresponding $R(z)$. In practice, (7.6) is implemented by solving the linear system of equations $P(\Delta t A)\mathbf{Y}^{m+1} = Q(\Delta t A)\mathbf{Y}^m$, where $P$ and $Q$ denote the numerator and denominator of $R$. When solving this system of equations the sparsity of $A$ is important.

The formula (7.6) should be compared with the corresponding expression for the true solution

$$\mathbf{Y}(t^{m+1}) = \exp(\Delta t B)\mathbf{Y}(t^m), \quad m = 0, 1, 2, \dots.$$

Such a comparison reveals that $R(z)$ should approximate $\exp(z)$; for instance with the Euler rule $R(z) = 1 + z$ consists of the first two terms of the expansion of $\exp(z)$ in powers of $z$. In general for a method of order $p$, $R(z)$ differs from $\exp(z)$ in terms of order $O(z^{p+1})$.

### 7.1.2 Fully discrete methods

As described above, the fully discrete scheme is obtained by time discretization of the time continuous problem (6.2). The fully discrete solution $\mathbf{U}^m$ at time $t^m$ is a vector $[U_0^m, U_1{,}^m, \dots, U_{M-1}^m]^T$, where $U_n^m$ is an approximation to $u(x_n, t^m)$. According to (7.6), the vectors $\mathbf{U}^m$ are recursively found from the formula

$$\mathbf{U}^{m+1} = R(\Delta t A)\mathbf{U}^m, \quad m = 0, 1, 2, \dots, \qquad (7.7)$$

where $A$ is the matrix in (6.2) and $R(z)$ the rational function of the specific time-stepping method being employed. The initial $\mathbf{U}^0$ is usually taken as in (6.4).

As an illustration we present the method corresponding to using the Euler rule (7.2) along with backward differences in space (see(6.3b)). When written componentwise, the formulae (7.7) become

$$U_n^{m+1} = U_n^m - \Delta t \frac{U_n^m - U_{n-1}^m}{\Delta x}, \quad n = 0, 1, \dots, M-1, \quad m = 0, 1, 2, \dots. \qquad (7.8)$$

### 7.1.3 Consistency, stability and convergence in the fully discrete case

The analysis of methods of the form (7.7) is also based on the ideas of consistency, stability and convergence. The *truncation errors* $\mathbf{L}^m$ are again defined by substituting the grid values of the theoretical solution in (7.7)

$$\Delta t \mathbf{L}_{m+1} = \mathbf{X}(u(\cdot, t^{m+1})) - R(\Delta t A)\mathbf{X}(u(\cdot, t^m)). \qquad (7.9)$$

Note the normalizing factor $\Delta t$ in the left-hand side. This is introduced because the format (7.7) does not directly approximate the partial differential equation (6.1), it rather approximates $\Delta t$ times (6.1). (Look at the example in (7.8): one would have to divide by $\Delta t$ before having a discrete analogue to (6.1).) Consistency of order $p$ in space and $q$ in time means that $\|\mathbf{L}^m\|$ behaves as $O(\Delta x^p + \Delta t^q)$ upon grid refinement. The discretization (7.8) has $p = q = 1$. Some authors define the truncation error to be the right-hand side of (7.9); for those authors the truncation error of (6.1) is $O(\Delta x \Delta t + \Delta t^2)$.

The discretization (7.7) is said to be *stable* (cf. (6.9)) if, for each finite time interval $0 \le t \le T$, a constant $C(T)$ can be found such that for all solutions of (7.7)

$$\|\mathbf{U}^m\| \le C(T)\|\mathbf{U}^0\|, \quad 0 \le t^m \le T.$$

Here it is crucial that $C(T)$ should not depend on $\Delta x$ and $\Delta t$. From (7.7), $\mathbf{U}^m = R(\Delta t A)^m \mathbf{U}^0$, so that the scheme is stable if and only if (cf. (6.10))

$$\|R(\Delta t A)^m\| \le C(T), \quad 0 \le t^m \le T. \qquad (7.10)$$

The global errors are now given by $\mathbf{U}^m - \mathbf{X}(u(\cdot, t^m))$ and a have a representation similar to (6.12), namely

$$\mathbf{E}^m = R(\Delta t A)^m \mathbf{E}^0 - \Delta t \sum_{\ell=1}^{m} R(\Delta t A)^{m-\ell}\mathbf{L}^\ell. \qquad (7.11)$$

From this formula it is easily concluded, as in Sect. 6.1.4, that a stable scheme with consistency of order $p$ in space and order $q$ in time has, for smooth solutions, global errors $\|\mathbf{E}^m\|$ that behave as $O(\Delta x^p + \Delta t^q)$.

**Exercise 38** Write componentwise the nine schemes resulting from combining (6.3a)-(6.3b) with (7.2)-(7.4). Study the consistency.

**Exercise 39** Prove (7.11).



## 7.2 The Von Neumann stability analysis

The stability condition (7.10) may be investigated by Fourier analysis. If $A$ is normal, then $R(\Delta t A)^m$ is also normal and (Sect. 1.2)

$$\|R(\Delta t A)^m\| = \rho(R(\Delta t A)^m) = \rho(R(\Delta t A))^m;$$

for stability this should be bounded by $C(T)$ whenever $0 \le t^m \le T$. It is not difficult to show (Richtmyer and Morton 1967) that this is equivalent to the existence of a constant $C'(T)$, independent of $\Delta x$ and $\Delta t$ such that for all $\Delta x$ and $\Delta t$

$$\rho(R(\Delta t A)) \le 1 + C'(T)\Delta t.$$

Upon recalling that the eigenvalues of $R(\Delta t A)$ are given by $R(\Delta t \sigma)$ with $\sigma$ eigenvalue of $A$, we conclude that stability is equivalent to

$$|R(\Delta t \sigma_n)| \le 1 + C'(T)\Delta t, \qquad (7.12)$$

as $\sigma_n$ runs through all the eigenvalues of $A$. These were found by Fourier analysis in Sect. 6.2. The condition (7.12) is the Von Neumann condition for fully discrete schemes.

Take (7.8) as an example and assume that the space and time grids are refined so as to keep the mesh ratio $r = \Delta t/\Delta x$ a constant. Here $R(z) = 1 + z$ and the eigenvalues of $A$ are given in (6.14). Therefore the stability requirement reads

$$\left|1 + r\left(\exp\left(-\frac{2\pi n i}{M}\right) - 1\right)\right| \le 1 + C'(T)\Delta t, \quad n = 0, 1, \ldots, M-1;$$

since the left-hand side is independent of $\Delta t$ this condition can only hold if

$$\left|1 + r\left(\exp\left(-\frac{2\pi i}{M}\right) - 1\right)\right| \le 1, \quad n = 0, 1, \ldots, M-1. \qquad (7.13)$$

It is easy to check that (7.13) is fulfilled if and only if $r \le 1$. Therefore the scheme is stable and convergent only if the ratio $r$ is kept below 1; this behavior is called ***conditional stability***.

**Exercise 40** Study the stability of the nine schemes introduced in Exercise 38.

## 7.3 The roles of stability and consistency from a Fourier viewpoint

The solution $\mathbf{U}^m$ given by the method (7.7) can be written as

$$\mathbf{U}^m = \sum_{n=-\infty}^{\infty} R(\Delta t \sigma_n)^m \hat{u}_n^0 \mathbf{X}(\phi_n), \quad m = 0, 1, 2, \ldots \qquad (7.14)$$

This is a fully discrete analogue of (6.18). Subtraction from (6.16) (evaluated at $t = t^m$) leads to the following expression for the global error

$$\mathbf{E}^m = \sum_{n=-\infty}^{\infty} [R(\Delta t\sigma_n)^m - \exp(\Delta t\mu_n)^m]\hat{u}_n^0 \mathbf{X}(\phi_n), \quad m = 0, 1, 2, \ldots.$$

This is a fully discrete version of (6.20) and may be analyzed as in Sect. 6.3.

For a consistent method, any fixed mode of the theoretical solution can be well approximated by choosing $\Delta x$ and $\Delta t$ suitably small. If $\Delta x$ is small, then, as we know, $\sigma_n$ is close to $\mu_n$. If $\Delta t$ is also small (relatively to $1/|\mu_n|$) then $R(\Delta t\sigma_n)$ is close to $\exp(\Delta t\mu_n)$ because $R(z)$ approximates $\exp(z)$ for $|z|$ small. Once more, the bad news is that for any given values of $\Delta x$ and $\Delta t$ there are values of $n$ for which $R(\Delta t\sigma_n)$ is a very poor approximation to $\exp(\Delta t\mu_n)$. For this reason consistency is not sufficient to guarantee convergence: one needs some control on the short wave length components (stability). As illustrated by the example in Sect. 7.2 stability may impose an upper bound on the value of $\Delta t$ that can be used on a given spatial mesh. It is often the case that this upper bound forces $\Delta t$ to be smaller (or even much smaller) than the value one would like to use for consistency reasons, i.e., the value that would make $\Delta t|\mu_n|$ small enough for the modes that are significantly present in the theoretical solution. This situation where stability rather than the natural time-scale of the theoretical solution dictates the choice of time step is called stiffness and unfortunately is a common occurrence in numerical differential equations (Hairer and Wanner 1991; Dekker and Verwer 1984). It is stiffness that makes implicit time stepping methods appealing for partial differential equations, explicit schemes at best lead to conditional stability (Sanz-Serna and Verwer 1989).

**Exercise 41** Prove that (7.14) is indeed the solution of (7.7) subject to (6.4).

### 7.4 Numerical dissipation and numerical dispersion

Let us particularize (7.14) to the scheme considered in Sect. 7.2 (backward differences in space and Euler time-stepping, $r = \Delta t/\Delta x$ a constant). The result is

$$\mathbf{U}^m = \sum_{n=-\infty}^{n} \{1 + r[\exp(-\kappa_n i\Delta x) - 1]\}^m \hat{u}_m^0 \mathbf{X}(\phi_n), \quad \kappa_n = 2\pi n/L. \quad (7.15)$$

We now wish to write the quantity in curly brackets as the exponential of its logarithm. As $\Delta x \to 0$ the Taylor expansion of this logarithm is

$$\begin{aligned}\log\{1 + r[\exp(-\kappa_n i\Delta x) - 1]\} &= \log\left\{1 - r\kappa_n i\Delta x - \frac{1}{2}r\kappa_n^2\Delta x^2 + \cdots\right\}\\ &= -r\kappa_n i\Delta x - \frac{1}{2}r(1-r)\kappa_n^2\Delta x^2 + \cdots\\ &= \Delta t\left(-\kappa_n i - \frac{1}{2}(1-r)\kappa_n^2\Delta x + \cdots\right).\end{aligned}$$

Thus, (7.15) may be rewritten as



$$\mathbf{U}^m = \sum_{n=-\infty}^{n} \exp(-\kappa_n i t^m)\exp\left(-\frac{1}{2}(1-r)\kappa_n^2\Delta x t^m + \cdots\right)\hat{u}_m^0\mathbf{X}(\phi_n).$$

Here the first exponential provides the correct advection with velocity 1; the second exponential arises from the error in the numerical method. For $r < 1$, the behavior of the leading, $O(\Delta x)$, term is *dissipative* (see Sect. 3.3). The dissipation coefficient is proportional to $\Delta x$ and, as expected, finer spatial grids induce less spurious dissipation. Note also that the dissipation coefficient decreases as $r$ approaches 1: unexpectedly, for this numerical scheme, longer time steps on a fixed spatial grid create less dissipation, see Figs. 11–12. For $r > 1$ we know from Sect. 7.2 that the scheme is unstable and hence useless; here we see that the numerical solution behaves as the solution to an ill posed backward heat equation. The case $r = 1$ is exceptional: in the formula above the $O(\Delta x)$ terms in the global error disappear. Indeed, for $r = 1$ all error terms disappear and the scheme provides the exact theoretical solution. This is possible because we are dealing with a very simple partial differential equation: the theoretical solution is constant along the characteristic lines $x = t + \xi$, and for $r = 1$ so is the numerical solution because, then from (7.8), $U_n^{m+1} = U_{n-1}^m$.

As we may see numerical methods have a dynamics of their own, which may be quite different from that of the equation being integrated (Sanz-Serna 1992). Modified equations are a popular way of analyzing spurious dissipation and dispersion (Warming and Hyett 1974; Griffiths and Sanz-Serna 1986).

**Exercise 42** Repeat the analysis in this section for the scheme for (6.1) based on central differences and trapezoidal time-stepping. Are the leading terms in the error dissipative or dispersive? Check your analysis by running a program.

## 8. The Practical Relevance of Fourier Analysis of Difference Methods

In Sects. 6–7 we have seen how Fourier techniques provide a powerful means for analyzing the stability and other properties of difference methods. The insights derived by Fourier analysis are essential when understanding difference methods. This insight, by itself, justifies the study of Fourier analysis.

The fly in the ointment is that, *strictly speaking*, Fourier analysis is only applicable under very restrictive conditions: periodic boundary conditions, linear equations, constant coefficients. This limitation in scope is particularly unfortunate because the difference numerical methods being analyzed have themselves no limitation in their range of applicability. Indeed one of the prime advantages of difference methods is their versatility: it is easy to write down a difference scheme for virtually any problem one may encounter.

In spite of these comments, Fourier techniques *are used in practice* in the analysis of linear problems with variable coefficients, of nonlinear problems and

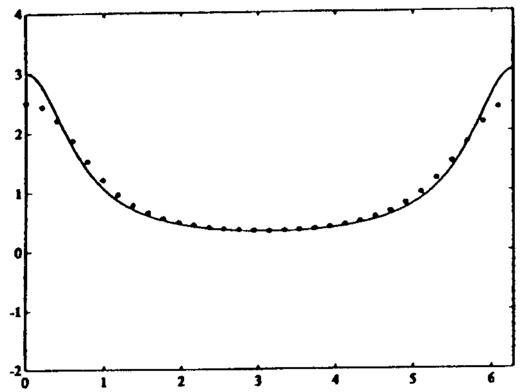


**Fig. 11.** The solution at time $t = 2\pi$ of the advection equation $\partial_t u = -\partial_x u$, on $0 \le x \le L = 2\pi$ for the initial datum in Exercise 8. The true solution is the solid line; this coincides with the initial condition because in $2\pi$ units of time the $u(\cdot, t)$ has moved $2\pi$ units to the right. The numerical method (7.8) is applied with $M = 32$ (so that $\Delta x = \pi/16$) and $\Delta t = 8\Delta x/9$ (so that 36 time steps have been needed). The dots represent the numerical solution; clearly the scheme is dissipative

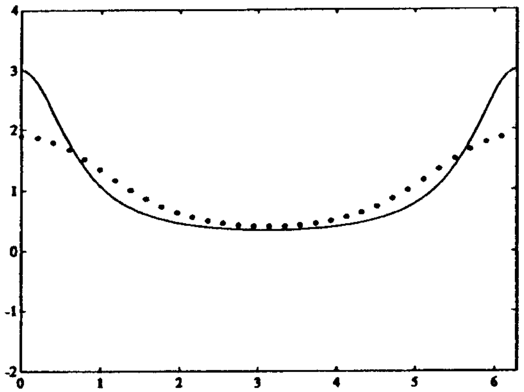


**Fig. 12.** The experiment in Fig. 11 has been repeated with the only change that now $\Delta t = \Delta x/4$. With this smaller time increment 128 time steps have to be taken to reach the final time $t = 2\pi$. In spite of the extra work, the result is more dissipative than that in Fig. 11

of problems whose boundary conditions are not periodic. For a linear problem with variable coefficients (say $\partial_t u = a(x,t)\partial_x u + b(x,t)u$) one 'freezes' the coefficients at a fixed, representative value ($a_0 = a(x_0,t_0)$, $b_0 = b(x_0,t_0)$) and analizes the resulting constant coefficient problem ($\partial_t u = a_0\partial_x u + b_0 u$). For a nonlinear problem one linearizes around a representative particular solution (for instance, for $\partial_t u = u\partial_x + \partial_{xx} u$, linearizing around $u = 0$ implies discarding the quadratic term $u\partial_x u$ to get $\partial_t u = \partial_{xx} u$). When the boundary conditions are not periodic, one changes during the analysis the boundary conditions into periodic conditions. In general (but there are some exceptions), such analyses are not mathematically justified. They nevertheless provide useful rule-of-thumb indications of the behavior of the scheme being employed. As expected, the predictions based on Fourier analysis tend to be optimistic. A Fourier analysis stability limit, such as $\Delta t \le (1/2)\Delta x^2$, is likely to be too generous; the real scheme will have a smaller stability limit due to nonlinear effects, boundary conditions etc. that are ignored in the analysis.



## 9. Spectral Methods

### 9.1 Spectral methods for periodic linear constant coefficient problems

#### 9.1.1 The Galerkin approach

For periodic, constant-coefficient, linear problems (3.1), the solution is available in closed form (3.6). Why do we then need difference methods for these problems? Well, we do not really need them; it is better to use (3.6) directly. It is then ironic that Fourier analysis of difference methods is mathematically justified precisely in those cases where the difference method is not really useful!

The only difficulty with (3.6) is that it comprises infinitely many terms; too many for a practical solution. We have to truncate somewhere and then the numerical solution is defined to be

$$u_G^N(x,t) = \sum_{n=-N}^{N} \exp(\mu_n t)\hat{u}_n^0 \phi_n(x). \qquad (9.1)$$

In $u_G^N$ the superscript $N$ indicates how many modes are being kept and G means Galerkin. Galerkin methods are projection methods and (9.1) is a projection method because

$$u_G^N(\cdot,t) = P_N(u(\cdot,t)); \qquad (9.2)$$

at each time $t$ the numerical solution is the orthogonal projection (see Exercise 2) of the theoretical solution onto the space of trigonometric polynomials of degree $\le N$. Hence the error equals $u(\cdot,t) - P_N(u(\cdot,t))$, which we know (Sect. 2.3) converges quickly to 0 if $u$ is smooth. How quickly depends on the exact smoothness; *the error may even be exponentially small as* $N \to \infty$.

To effectively construct (9.1) we need the Fourier coefficients $\hat{u}_n^0$, with $|n| \le N$; these would have to be calculated by numerical evaluation of the integrals (2.5).

The method (9.1) is said to be a *spectral Galerkin method*. It uses the explicit knowledge of the spectrum of eigenvalues of the operator $P(\partial_x)$ in the equation (3.1) being solved.

### 9.1.2 The collocation approach

One could avoid the computation of the $\hat{u}_n^0$ in (9.1) if these were replaced by the discrete Fourier coefficients $\tilde{u}_n^0$ of $u^0$ relative to a grid with $M = 2N$ points. The result would be an *alternative* numerical *spectral* approximation given by

$$u_\psi^N(x,t) = \sum_{n=-N}^{N}{}'' \exp(\mu_n t)\tilde{u}_n^0 \phi_n(x). \tag{9.3}$$

The subscript $\psi$ means *pseudospectral* and indeed (9.3) is called a pseudospectral solution or, alternatively, a spectral collocation solution. Now the $\tilde{u}_n^0$ are found by discrete Fourier transform as explained in Sect. 5.2.2. The cost of forming (9.3) is then $O(M \log_2 M)$ operations, which is competitive with finite difference methods. Without the fast transform, (9.3) would require $O(M^2)$ operations, which is not really so appealing.

As we discussed, (9.1) is based on projections: one projects $u^0$ to find the required coefficients and the method gives back the projection of the theoretical solution, see (9.2). In the pseudospectral solution (9.3) one interpolates $u^0$ on the grid. However the method does not yield at time $t$ the interpolant of $u(\cdot,t)$. That would be too much; it would mean that the method was actually exact at the grid points. To clarify this, recall that, by (5.13),

$$\tilde{u}_n^0 = \sum_{m=-\infty}^{\infty} \hat{u}_{n+mM}^0$$

and hence

$$u_\psi^N(x,t) = \sum_{n=-\infty}^{\infty} E_n(t)\hat{u}_n^0 \phi(x)$$

where

$$E_n(t) = \exp(\mu_m t),$$

if $n \equiv m$, $|m| < N$ (i.e., $m$ is the mode below the Nyquist limit into which the mode $n$ becomes aliased), and

$$E_n(t) = \frac{1}{2}\exp(\mu_N t) + \frac{1}{2}\exp(\mu_{-N} t),$$

if $n \equiv N$ (i.e., $n$ is aliased into the Nyquist limit).

Comparing with the theoretical solution (3.6), we see that the source of error in (9.3) is the following. High wave numbers are aliased when interpolating $u^0$ and later are evolved with exponentials $\exp(\mu_m t)$ that really correspond to the modes into which they have been aliased. High wave numbers are kept in the numerical solution but falsified. On the Galerkin solution they are just suppressed, which is in a way another form of falsification.

In any case, both the Galerkin and pseudospectral solution approximate without error all the modes representable in the grid (cf. Fig. 10). With these methods the errors depend on the high Fourier coefficients $\hat{u}_n^0$, $|n| > N$. For this reason, the accuracy of $u_G$ or $u_\psi$ depends only on the decay of the $\hat{u}_n^0$ and therefore is only limited by the smoothness of $u^0$. The smoother $u^0$ the faster the convergence. If $u^0$ is very regular, the errors may be exponentially small as $N \to \infty$. For a stable time-continuous difference scheme of order of consistency $O(\Delta x^p)$, errors are never better than $O(\Delta x^p)$ for very smooth solutions and may be worse than that if $u$ is not so smooth.



**Exercise 43** Use (9.3) to solve the heat equation. Write a program that, for a given initial $u^0$, plots the numerical solution at any given time.

## 9.2 Pseudospectral difference matrices

### 9.2.1 First derivatives

The spectral methods introduced in Sect. 9.1 for periodic, constant coefficient, linear problems can be extended to periodic, variable coefficient, linear problems and even to periodic, nonlinear problems. This extension is presented in Sect. 9.3. We now make a small detour. We wish to study the following problem. We are given the grid values $\mathbf{X}(f)$ of an $L$-periodic function $f(x)$ and are asked to find approximations to the grid values $\mathbf{X}(f')$ of the derivative. We could of course use standard finite-difference formulae, such as $(f(x_{n+1}) - f(x_n))/\Delta x$, but such standard formulae do not take into account that we are dealing with periodic functions. A better recipe is: (i) interpolate the given grid values as in (5.6), (ii) differentiate the interpolant $I_N$ to find, in view of (2.12),

$$I_N(f)' = \sum_{n=-N}^{N}{}'' \lambda_n \tilde{f}_n \phi_n(x),$$

and (iii) evaluate the derivative at the grid points. The end result is a vector $\mathbf{X}'(f)$ of approximations to the values of $f'$ at grid points (not to be confused with the vector of exact derivative values $\mathbf{X}(f')$)

$$\mathbf{X}'(f) = \sum_{n=-(N-1)}^{N-1} \lambda_n \tilde{f}_n \mathbf{X}(\phi_n). \tag{9.4}$$

Note that the terms with $|n| = N$ do not feature in (9.4). Why? Recall from (5.6) that $\tilde{f}_{-N} = \tilde{f}_N$. On the other hand, $\lambda_{-N} = -\lambda_N$ and then the $|n| = N$ terms in $I_N(f)'$ equal $\lambda_N \tilde{f}_N[\phi_N(x) - \phi_{-N}(x)]$, i.e., $2i\lambda_N \tilde{f}_N \sin(2\pi N x/L)$; the function $\sin(2\pi N x/L)$ has zero grid values and therefore the $|n| = N$ terms do not contribute to (9.4).

The process of finding $\mathbf{X}'(f)$ as a function of the data $\mathbf{X}(f)$ is linear and therefore there exists an $M \times M$ matrix $D$ such that

$$\mathbf{X}'(f) = D\mathbf{X}(f).$$

This matrix is called the *pseudospectral differentiation matrix*. It is a real matrix: for real $\mathbf{X}(f)$, (9.4) contains the complex conjugate of each of its terms.

The following representation is most important

$$D = \left(P\frac{1}{M}F_M\right)^{-1} \Lambda \left(P\frac{1}{M}F_M\right). \tag{9.5}$$

Here $P$ is the $M \times M$ permutation matrix

$$P = \begin{bmatrix} O_N & I_N \\ I_N & O_N \end{bmatrix}$$

and $\Lambda$ is an $M \times M$ diagonal matrix

$$\Lambda = \mathrm{Diag}\,(0, \lambda_{-N+1}, \lambda_{-N+2}, \dots, \lambda_{-1}, \lambda_0, \lambda_1, \dots, \lambda_{N-2}, \lambda_{N-1}).$$

In (9.5), $D$ is written as the product of three factors. The rightmost factor comprises the matrix $(1/M)F_M$ that finds the discrete Fourier coefficients as in (5.5) and the permutation matrix $P$ that rearranges them as in (5.8). The central matrix $\Lambda$ multiplies each discrete Fourier coefficient by the corresponding eigenvalue $\lambda_n$. Note that $\Lambda$ includes a 0 entry that suppresses the $|n| = N$ contribution. The leftmost factor in the right-hand side of (9.5) undoes the action of the rightmost factor, i.e., computes grid values given discrete Fourier coefficients.

The Fourier matrix $F_M$ has all its entries $\neq 0$ and, as a consequence, the matrix $D$ is not a sparse matrix: $D$ is full. When using finite differences, differentiation is performed through a sparse matrix, see e.g. (6.3). For standard central differences (6.3c), the differentiation matrix has two nonzero entries per row. For fourth-order differencing (Exercise 31), sixth-order differencing, ... the number of nonzero entries per row is four, six, ...; higher order implies less sparsity. The pseudospectral matrix can be seen (Fornberg 1975, 1987, 1990) as the full limit of difference matrices of increasing order of accuracy and decreasing sparsity.

Since $P$ is its own inverse, (9.5) may be rewritten as

$$D = F_M^{-1} P \Lambda P F_M. \tag{9.6}$$

To find $D\mathbf{X}$ for a given vector $\mathbf{X}$ requires (i) an FFT, (ii) a permutation (in MATLAB, this is achieved by the function **fftshift**, see Sect. 5.2.2), (iii) $M$ complex multiplications by the diagonal entries of $\Lambda$, (iiii) another permutation and (v) an inverse FFT transform. This involves a computational cost that is essentially $O(M)$. I emphasize that it is not advisable to find explicitly $D$ by multiplying the five matrices in (9.6): if one uses the explicit form of $D$ and carries out the multiplication $D\mathbf{X}$ by the standard matrix-times-vector recipe, then the work is $O(M^2)$.



### 9.2.2 Higher derivatives

The idea in Sect. 9.2.1 is readily generalized to higher derivatives. To obtain approximations to the grid values of $f''$, we again differentiate the interpolant to get

$$I_N(f)'' = \sum_{n=-N}^{N}{}'' \lambda_n^2 \tilde{f}_n \phi_n(x),$$

and then evaluate at grid points

$$\mathbf{X}''(f) = \sum_{n=-N}^{N}{}'' \lambda_n^2 \tilde{f}_n \mathbf{X}(\phi_n). \tag{9.7}$$

Now the $|n| = N$ terms do not disappear because $\lambda_{-N}^2 = \lambda_N^2$. In matrix form

$$\mathbf{X}''(f) = D^{(2)}\mathbf{X}(f),$$

with

$$D^{(2)} = F_M^{-1} P \Lambda^{(2)} P F_M,$$

where, in turn,

$$\Lambda^{(2)} = \mathrm{Diag}\,(\lambda_{-N}^2, \lambda_{-N+1}^2, \lambda_{-N+2}, \dots, \lambda_{-1}^2, \lambda_0^2, \lambda_1^2, \dots, \lambda_{N-2}^2, \lambda_{N-1}^2).$$

A small point: $\Lambda^{(2)}$ is not the square of $\Lambda$; the $(1,1)$ entry of $\Lambda^2$ is zero. For this reason

$$D^2 = (F_M^{-1} P \Lambda P F)\,(F_M^{-1} P \Lambda P F_M) = F_M^{-1} P \Lambda^2 P F_M$$

does not coincide with $D^{(2)}$. If you first interpolate and then differentiate twice you get a contribution with $|n| = N$ (in the trigonometric representation of the interpolant as in Sect. 5.2.4, the second derivative of $\cos(2\pi N x/L)$ is a multiple of $\cos(2\pi N x/L)$ and stays). If you interpolate, differentiate once, evaluate, and again interpolate and differentiate, the contribution with $|n| = N$ perishes in the first differentiation and never raises from the dead. Nevertheless, the difference between $D^2$ and $D^{(2)}$ is in practice very small: these two matrices only differ in the way they treat the highest discrete Fourier mode.

Higher derivative matrices $D^{(k)}$ can be constructed in a similar way. Alternatively these may be replaced by powers $D^k$ of the first-derivative matrix $D$. Independently of the value of $k$ the multiplication $D^{(k)}\mathbf{X}$ (or $D^k\mathbf{X}$), requires a direct and an inverse FFT plus $M$ multiplications.

**Exercise 44** Prove that $D$ is a skew-symmetric matrix. Prove that $D$ is a circulant matrix.

**Exercise 45** Use (9.6) to find explicitly $D$ when $M = 4$. Check that, for the matrix $D_4$ you have found, $D_4\mathbf{X}(f)$ provides the correct nodal values of the derivative of $f$ when $f$ is one of the functions 1, $\cos(2\pi x/L)$, $\sin(2\pi x/L)$,

$\cos(4\pi x/L)$. What is the situation when $f(x) = \sin(4\pi x/L)$? Find $D_4\mathbf{X}(f)$ for $f(x) = \cos(2\pi nx/L)$ or $f(x) = \sin(2\pi nx/L)$ and explain the results in terms of aliasing. Compare the entries in $D_4$ with those of the $4 \times 4$ matrix that arises from standard central differences (this is the negative of the matrix in (6.3c)).

## 9.3 The pseudospectral method for periodic nonlinear problems

Both the Galerkin and pseudospectral methods presented in Sect. 9.1 can be extended outside the class of periodic, constant coefficient, linear problems (Canuto et al. 1988; Gottlieb and Orszag 1977). The extension is much easier for the pseudospectral case and we therefore only consider pseudospectral methods.

To see how this extension, first suggested by Kreiss and Oliger in 1972, works, let us first look at the example of the heat equation $\partial_t u = \partial_{xx} u$. The pseudospectral solution (9.3) is

$$u_\psi^N(x,t) = \sum_{n=-N}^{N}{}'' \exp(\lambda_n^2 t)\tilde{u}_n^0 \phi_n(x).$$

Denote by $\mathbf{U}(t)$ the $M$-vector of grid values of $u_\psi^N$ at time $t$. Obviously

$$\mathbf{U}(t) = \sum_{n=-N}^{N}{}'' \exp(\lambda_n^2 t)\tilde{u}_n^0 \mathbf{X}(\phi_n) \tag{9.8}$$

and

$$\frac{d}{dt}\mathbf{U}(t) = \sum_{n=-N}^{N}{}'' \lambda_n^2 \exp(\lambda_n^2 t)\tilde{u}_n^0 \mathbf{X}(\phi_n). \tag{9.9}$$

Now, from (9.8), $\mathbf{U}(t)$ has discrete Fourier coefficients $\exp(\lambda_n^2 t)\tilde{u}_n^0$, so that comparing the right-hand side of (9.9) with (9.7) we see that this right-hand side is none other than $D^{(2)}\mathbf{U}(t)$. Therefore, (9.9) may be written in a simple form:

$$\frac{d}{dt}\mathbf{U}(t) = D^{(2)}\mathbf{U}(t).$$

This is very similar to a time-continuous finite difference scheme of the format (6.2); the only difference is that instead of a (sparse) finite-difference matrix we now have a (full) pseudospectral difference matrix. The pseudospectral matrix differentiates exactly all Fourier modes below the Nyquist limit. As the grid is refined more modes are differentiated without error.

This example gives the key to writing pseudospectral methods for problems that, *while being periodic*, have variable coefficients or are nonlinear. One considers as unknown a vector $\mathbf{U}(t)$ of grid values and one writes a system of differential equations

$$(d/dt\mathbf{U}(t)) = \mathbf{F}(t, \mathbf{U}(t)), \tag{9.10}$$



for $\mathbf{U}(t)$. The right-hand side function $\mathbf{F}$ is constructed from the partial differential equation being solved by replacing the derivative operators $\partial_x^k$ by the matrices $D^{(k)}$ (or $D^k$). For instance, for the Korteweg-de Vries equation,

$$\partial_t u = -3\partial_x u^2 - \partial_{xxx} u,$$

a pseudospectral scheme (Frutos and Sanz-Serna 1992) would have $\mathbf{F}$ given by the nonlinear function

$$\mathbf{F}(\mathbf{V}) = -3D\mathbf{V}^2 - D^3\mathbf{V},$$

where the vector $\mathbf{V}^2$ is obtained by squaring the entries of $\mathbf{V}$.

In (9.10) the variable $t$ is still continuous so that one has to integrate in time by some numerical method for ordinary differential equations (Sect. 7.1.1). An automatic package from a mathematical software library may be a sensible choice, but one also may consider some home-made algorithm. In any case one should be careful because (9.10) is both stiff and full. With any time integration method one may chose, one needs to be able to write a subroutine that evaluates the right-hand side function $\mathbf{F}$ at any given vector $\mathbf{V}$. In the Korteweg-de Vries example, the multiplications by $D^3$ and $D$ are carried out by FFT as discussed in Sect. 9.2.1. The right-most $F_M^{-1}$ implicit in $D$ (see (9.6)) and in $D^3$ can be taken as a common factor. Then the evaluation of $\mathbf{F}$ requires two transforms and an inverse transform, plus $M$ multiplications to find the entries of $\mathbf{V}^2$ and $2M$ additional multiplications by the diagonal entries of $\Lambda$ and $\Lambda^3$.

In practice it may be better not to use the system (9.10) directly. One rather performs the time integration in terms of the transformed vector $\tilde{\mathbf{U}}(t) = PF_M\mathbf{U}(t)$, whose entries are, except for a normalizing factor $M$, the discrete Fourier coefficients of the solution. From (9.10), the transformed vector satisfies the differential system

$$\frac{d}{dt}\tilde{\mathbf{U}}(t) = \tilde{\mathbf{F}}(t, \tilde{\mathbf{U}}(t)),$$

with

$$\tilde{\mathbf{F}}(\tilde{\mathbf{V}}) = PF_M\mathbf{F}(F_M^{-1}P\tilde{\mathbf{V}}).$$

In the Korteweg-de Vries example, the new right-hand side function $\tilde{\mathbf{F}}$ is

$$\begin{aligned} PF_M\mathbf{F}(F_M^{-1}P\tilde{\mathbf{V}}) &= PF_M\left(-3D\mathbf{V}^2 - D^3\mathbf{V}\right), \\ &= -3\Lambda PF_M\mathbf{V}^2 - \Lambda^3\tilde{\mathbf{V}} \end{aligned}$$

where

$$\mathbf{V} = F_M^{-1}P\tilde{\mathbf{V}}. \tag{9.11}$$

Therefore when the system is written in terms of the transformed vector, an evaluation of the right-hand side function $\tilde{\mathbf{F}}$ demands an inverse Fourier transform (9.11) to find the nodal values $\mathbf{V}$ from discrete Fourier coefficients, $M$ multiplications to find the entries of $\mathbf{V}^2$, a further discrete Fourier transform to find $F_M\mathbf{V}^2$ and the $2M$ multiplications by the diagonal entries of $\Lambda$ and $\Lambda^3$. This saves a transform when compared with the evaluation of $\mathbf{F}$.

The main limitation of the Fourier pseudospectral approach is the restriction to periodic boundary conditions. Finite differences may cope successfully with any boundary condition. Pseudospectral methods based on polynomial rather than trigonometric basis functions of course exist (Canuto et al. 1988; Gottlieb and Orszag 1977) and can deal with nonperiodic boundary conditions.

The good news is that the rate of convergence of Fourier pseudospectral methods is only restricted by the smoothness of the solution $u$. For very smooth solutions errors are exponentially small as $\Delta x \to 0$. This is to be compared with the situation for finite differences where the error is never better than $O(\Delta x^p)$, with $p$ determined by the specific scheme being used.

In my experience (Abia and Sanz-Serna 1990; Frutos et at. 1990, 1991; Frutos and Sanz-Serna 1989, 1992) the superiority of spectral methods over finite differences or finite elements is dramatic. Let me report an experiment (Abia and Sanz-Serna 1990). The equation being integrated is nonlinear and describes waves in a fluidized bed. A pseudospectral method, with accurate time integration, yielded an error of $6 \times 10^{-3}$ when $M = 4$. The error is reduced down to $1 \times 10^{-5}$ when $M$ is doubled ($M = 8$). This is an error reduction by a factor 600! A second order finite-difference scheme had with $M = 32, 64, 128$ errors of $3 \times 10^{-2}$, $7 \times 10^{-3}$. $1 \times 10^{-3}$. Here the error is only divided by a factor of about 4 when $M$ is doubled. Note also that finite differences with $M = 128$ (CPU time 13 seconds) are 100 times less accurate than the pseudospectral method with $M = 8$ (CPU time 7 seconds).

**Exercise 46** Write a program for the pseudospectral method for the Korteweg-de Vries equation. Use the classical Runge-Kutta formula for the integration in time (Lambert 1991). Write a program based on finite differences and compare with the pseudospectral method.

I am thankful to J. de Frutos who helped me with the preparation of the manuscript. I have been supported by grant DGICYT PB92-254.

## References

Abia, L. and Sanz-Serna, J.M. (1990): Computing **44** 187

Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. (1988): *Spectral Methods in Fluid Dynamics* (Springer, New York)

Cooley, J. W. and Tukey, J. W. (1965): Math. Comput. **19** 297

Dekker, K. and Verwer, J.G. (1984): *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations* (North Holland, Amsterdam)

Fornberg, B. (1975): SIAM J. Numer. Anal. **12** 509

Fornberg, B. (1987): Geophysics **52** 483

Fornberg, B. (1990): SIAM J. Numer. Anal. **27** 904



Frutos, J. de, Ortega, T. and Sanz-Serna, J.M. (1990): Comput. Methods Appl. Mech. Eng. **80** 417

Frutos, J. de, Ortega, T. and Sanz-Serna, J.M. (1991): Math. Comput. **57** 109

Frutos, J. de and Sanz-Serna, J.M. (1989): J. Comput. Phys. **83** 407

Frutos, J. de and Sanz-Serna, J.M. (1992): J. Comput. Phys. **103** 160

Golub, G.H. and Van Loan, C.F. (1989): *Matrix Computations* (John Hopkins University Press, Baltimore)

Gottlieb, D. and Orszag, S.A. (1977): *Numerical Analysis of Spectral Methods: Theory and Applications* (SIAM-CBMS, Philadelphia)

Griffiths, D.F. and Sanz-Serna, J.M. (1986): SIAM J. Sci. Stat. Comput. **7** 994

Hairer, E., Nørsett, S.P. and Wanner, G. (1993): *Solving Ordinary Differential Equations I, Nonstiff Problems* (Springer, Berlin)

Hairer, E. and Wanner, G. (1991): *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems* (Springer, Berlin)

Hamming, R.W. (1973): *Numerical Methods for Scientists and Engineers* (McGraw-Hill, New York)

Horn, R.A. and Johnson, C.R. (1985): *Matrix Analysis* (Cambridge University Press, Cambridge)

Kreiss, H. and Oliger, J. (1972): Tellus **24** 199

Kreiss, H. and Oliger, J. (1973): *Methods for the Approximate Solution of Time Dependent Problems* Global Atmospheric Research Program Publications Series No, 10 (World Meteorological Organization, Geneva)

Lambert, J.D. (1991): *Numerical Methods for Ordinary Differential Equations, The Initial Value Problem* (Wiley, Chichester)

Mitchell, A.R. and Griffiths, D.F. (1980): *The Finite Difference Method in Partial Differential Equations* (Wiley, Chichester)

Press W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1989): *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press, Cambridge)

Richtmyer R.D. and Morton, K.W. (1967): *Difference Methods for Initial-Value Problems* (Wiley-Interscience, New York)

Sanz-Serna, J.M. (1985): "Stability and Convergence in Numerical Analysis, a Simple, Comprehensive Account", in Nonlinear Differential Equations ed. by Hale, J.K. and Martínez-Amores, P. (Pitman, Boston) pp. 64–113

Sanz-Serna, J.M. (1991): "Two Topics in Nonlinear Stability", in Advances in Numerical Analysis, ed. by W. Light (Clarendon Press, Oxford), pp. 147-174.

Sanz-Serna, J.M. (1992): "Numerical Ordinary Differential Equations vs. Dynamical Systems", in The Dynamics of Numerics and the Numerics of Dynamics, ed. by Broomhead, D.S. and Iserles, A. (Clarendon Press, Oxford) pp. 81–106.

Sanz-Serna, J.M. and Calvo, M.P. (1994): *Numerical Hamiltonian Problems* (Chapman & Hall, London)

Sanz-Serna, J.M. and Palencia, C. (1985): Math. Comput. **45** 143

Sanz-Serna, J.M. and Verwer, J.G. (1989): Appl. Numer. Math. **5** 117

Strang, G. (1986): *Introduction to Applied Mathematics* (Wellesley-Cambridge Press, Wellesley)

Strang, G. and Fix, G.J. (1973): *An Analysis of the Finite Element Method* (Prentice-Hall, Englewood Cliffs)
Tadmor E. (1986): SIAM J. Numer. Anal. **23** 1
Trefethen, LL. N. (1982): SIAM Review **24** 113
Trefethen, LL. N. (1983): J. Comput. Phys. **49** 199
Vichnevetsky, R. (1987a): Appl. Numer. Math. **3** 133
Vichnevetsky, R. (1987b): Int. J. Numer. Methods Fluids **7** 409
Vichnevetsky, R. (1989): Int. J. Numer. Methods Fluids **9** 623
Vichnevetsky, R. (1990): Computers Math. Applics. **19** 59
Vichnevetsky, R. (1992): Appl. Numer. Math. **10** 195
Warming, R.F. and Hyett, B.J. (1974): J. Comput. Phys. **14** 159
Whitham, G.B. (1974): *Linear and Nonlinear Waves* (Wiley, New York)

# A Fast Algorithm for the Generation of Random Numbers with Exponential and Normal Distributions

Julio F. Fernández[1,2] and Juan Rivero[2]

[1] Instituto Carlos I de Física Teórica y Computacional and Departamento de Física Aplicada, Universidad de Granada, 18071-Granada, Spain.
[2] Instituto Venezolano de Investigaciones Científicas, Apartado 21827, Caracas 1020A, Venezuela

**Abstract:** Algorithms for the generation of pseudorandom numbers with normal and exponential distributions are described here. No transcendental functions need to be evaluated; furthermore, only two uniform deviates per generation are required; no tables are used. These algorithms are much faster than other exponential and normal random number generators.

## 1. Introduction

Many simulations in computational physics require random numbers with a given distribution. Exponential and normal distributions are often needed. Existing random number generators are often inaccurate and they are time consuming. It has been shown recently that the inaccuracy issue can sometimes be reasonably serious [1]. Exponential and normally distributed random numbers are generated particularly slowly because one or more transcendental functions and/or several uniform deviates must be evaluated for each random number generated [2,3]. New algorithms for the generation of pseudorandom numbers with normal and exponential distributions are described in this lecture. No transcendental functions need to be evaluated; furthermore, only two uniform deviates per generation are required. Its accuracy is easy to control. No tables are used. It is much faster than other generators.

As part of an introduction for students who are unfamiliar with this subject, one of the simplest methods to generate exponentially distributed random numbers is explained next. Take a random number $x$ (supplied, for instance by a built in generator in your computer), distributed uniformly in the interval $(0,1)$, that is, $P(x)=1$, for $0<x<1$, and $P(x)=0$, for $x<0$ and $x>1$. There is a function $y(x)$ that transforms the *uniform deviate* $x$ into the desired *exponential deviate* $y$; it must fulfill

$$P(x)|dx| = P(y)|dy| \tag{1.1}$$