3
Chebyshev method

The Fourier method is appropriate for periodic problems, but is not adapted to nonperiodic problems because of the existence of the Gibbs phenomenon at the boundaries. In the case of nonperiodic problems, it is advisable to have recourse to better-suited basis functions. Orthogonal polynomials, like Chebyshev polynomials, constitute a proper alternative to the Fourier basis. The Chebyshev series expansion may be seen as a cosine Fourier series, so that it possesses the valuable properties of the latter concerning, in particular, the convergence and the possible use of the FFT. On the other hand, the Chebyshev series expansion is exempt from the Gibbs phenomenon at the boundaries.

Another possible choice of an orthogonal polynomial basis is constituted by the Legendre polynomials. These polynomials share a number of properties with the Chebyshev polynomials. They present some advantages concerning the properties of the discrete operators and the numerical quadrature. On the other hand, no fast transform algorithm is known for Legendre polynomials. Only Chebyshev polynomials are discussed in this book, but the methods and algorithms described also apply to Legendre polynomials with only technical changes required by their specific properties. We refer to the books by Gottlieb and Orszag (1977), Canuto *et al.* (1988), Bernardi and Maday (1992) or Funaro (1992) for discussions on the properties and applications of the Legendre polynomials.

The present chapter is intended to give a general view of Chebyshev polynomials and their applications to the solution of boundary value problems. Two classical approaches, Galerkin-type (tau method) and collocation, will be addressed. In this latter case, it will be pointed out that the Chebyshev

truncated series expansion can be seen as the Lagrange interpolation polynomial based on the collocation points. Direct and iterative methods for solving the algebraic systems, resulting from the Chebyshev approximation, will be described.

3.1 Generalities on Chebyshev polynomials

The Chebyshev polynomial of the first kind $T_k(x)$ is the polynomial of degree k defined for $x \in [-1, 1]$ by

$$T_k(x) = \cos\left(k \cos^{-1} x\right) , k = 0, 1, 2, \dots , \tag{3.1}$$

therefore, $-1 \leq T_k \leq 1$. By setting $x = \cos z$, we have

$$T_k = \cos kz , \tag{3.2}$$

from which it is easy to deduce the expressions for the first Chebyshev polynomials

$$T_0 = 1 , \;\; T_1 = \cos z = x , \;\; T_2 = \cos 2z = 2 \cos^2 z - 1 = 2\, x^2 - 1 , \dots .$$

More generally, from the Moivre formula, we get

$$\cos kz = \mathcal{R}e \left\{ \left(\cos z + \underline{i} \sin z\right)^k \right\}$$

and then, by application of the binomial formula, we may express the polynomial T_k as

$$T_k = \frac{k}{2} \sum_{m=0}^{[k/2]} (-1)^m \, \frac{(k-m-1)!}{m!\; (k-2\,m)!} \, (2\,x)^{k-2\,m} , \tag{3.3}$$

where $[\phi]$ denotes the integer part of ϕ.

From the trigonometrical identity

$$\cos\left(k+1\right) z + \cos\left(k-1\right) z = 2 \cos z \, \cos kz$$

we deduce the recurrence relationship

$$T_{k+1} - 2\, x\, T_k + T_{k-1} = 0 , \;\; k \geq 1 , \tag{3.4}$$

which allows us, in particular, to deduce the expression of the polynomials T_k, $k \geq 2$, from the knowledge of T_0 and T_1. The graph of the first polynomials is shown in Fig. 3.1.

Expression (3.3) may be useful in some special circumstances but the representation (3.2) is generally used in computational as well as theoretical studies.

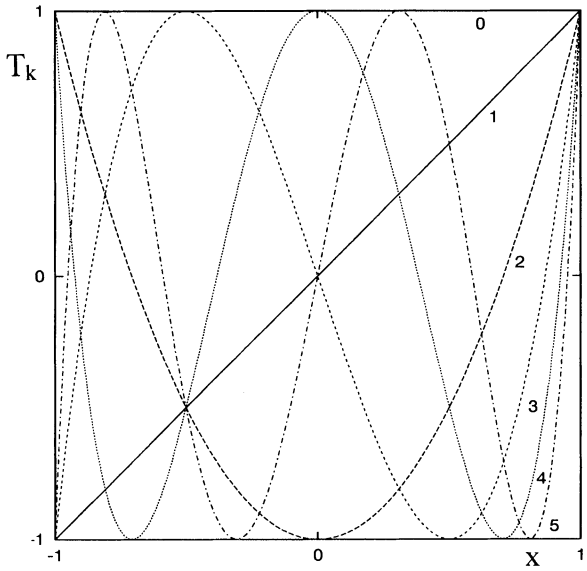

FIGURE 3.1. Graphs of the first Chebyshev polynomials, $T_k(x)$, for $k = 0, \cdots, 5$.

Now we list some properties useful for the understanding and application of Chebyshev polynomials to the solution of ordinary or partial differential equations ; other properties will be discussed in the sequel when necessary. And then, a richer set of formulas is given in Appendix A.

The values of T_k and its first-order derivative T_k' at $x = \pm 1$ are given by

$$T_k(\pm 1) = (\pm 1)^k \ , \quad T_k'(\pm 1) = (\pm 1)^{k+1} k^2 \, . \tag{3.5}$$

The knowledge of these values can be of interest when prescribing boundary conditions. It is important to note that

$$T_k(-x) = (-1)^k T_k(x) \, , \tag{3.6}$$

that is the parity of the polynomial is the same as its degree k.

The polynomial T_k vanishes at the points x_i (Gauss points) defined by

$$x_i = \cos\left(i + \frac{1}{2}\right)\frac{\pi}{k} \, , \quad i = 0, \ldots, k-1 \tag{3.7}$$

and it reaches its extremal values ± 1 at the points x_i (Gauss-Lobatto points) defined by

$$x_i = \cos\frac{\pi\, i}{k} \, , i = 0, \ldots, k \, . \tag{3.8}$$

Note that such points are the zeros of the polynomial $\left(1 - x^2\right) T_k'(x)$.

A recurrence relation on the derivative can easily be obtained. First, the differentiation of T_k gives

$$T'_k = \frac{d}{dz}(\cos kz)\frac{dz}{dx} = k\frac{\sin kz}{\sin z}\,,$$

where we have used the representation (3.2). Then, by the application of trigonometrical formulas, we get the relation

$$\frac{T'_{k+1}}{k+1} - \frac{T'_{k-1}}{k-1} = 2\,T_k \tag{3.9}$$

valid for $k > 1$. A similar formula for the pth derivative is obtained by successive differentiations of (3.9).

The Chebyshev polynomials are orthogonal on $[-1, 1]$ with the weight

$$w = \left(1 - x^2\right)^{-1/2}\,. \tag{3.10}$$

Let the scalar product be

$$(u, v)_w = \int_{-1}^{1} u\,v\,w\,dx\,, \tag{3.11}$$

so that the orthogonality property is

$$(T_k, T_l)_w = \int_{-1}^{1} T_k\,T_l\,w\,dx = \frac{\pi}{2}c_k\,\delta_{k,l}\,, \tag{3.12}$$

where $\delta_{k,l}$ is the Kronecker delta and c_k is defined by

$$c_k = \begin{cases} 2 & \text{if } k = 0\,, \\ 1 & \text{if } k \geq 1\,. \end{cases} \tag{3.13}$$

The Chebyshev approximation makes extensive use of the Gauss quadrature formulas. For the Gauss-Lobatto points $x_i = \cos \pi\, i/N$, $i = 0, \ldots, N$ [see Eq.(3.8)], generally used in collocation methods, the quadrature formula applied to any function $p(x)$ gives

$$\int_{-1}^{1} p\,w\,dx \cong \frac{\pi}{N}\sum_{i=0}^{N}\frac{p(x_i)}{\overline{c}_i}\,, \tag{3.14}$$

where

$$\overline{c}_k = \begin{cases} 2 & \text{if} \quad k = 0\,, \\ 1 & \text{if} \quad 1 \leq k \leq N-1\,, \\ 2 & \text{if} \quad k = N\,. \end{cases} \tag{3.15}$$

The relation (3.14) is exact if $p(x)$ is a polynomial of degree $2N - 1$ at most [see Mercier (1989) for the proof and formulas associated with other sets of points].

From Eq.(3.14) we may derive the discrete orthogonality relation based on the Gauss-Lobatto points x_i, $i = 0, \ldots, N$. For $k \neq N$ or $l \neq N$, the use of (3.14) gives an exact approximation to the integral in (3.12) since $T_k T_l$ is a polynomial of degree at most $2N - 1$:

$$\frac{\pi}{2} c_k \, \delta_{k,l} = \int_{-1}^{1} T_k \, T_l \, w \, dx = \frac{\pi}{N} \sum_{i=0}^{N} \frac{1}{\overline{c}_i} T_k(x_i) \, T_l(x_i) \, .$$

For $k = l = N$, this last formula remains exact provided c_k in the left-hand side is replaced by $\overline{c}_N$ $(= 2)$. Therefore, the discrete orthogonality relation is

$$\sum_{i=0}^{N} \frac{1}{\overline{c}_i} T_k(x_i) \, T_l(x_i) = \frac{\overline{c}_k}{2} \, N \, \delta_{k,l} \tag{3.16}$$

valid for $0 \leq k, l \leq N$.

3.2 Truncated Chebyshev series

3.2.1 Calculation of Chebyshev coefficients

Let us consider the Chebyshev approximation of the function $u(x)$ defined for $x \in [-1, 1]$:

$$u_N(x) = \sum_{k=0}^{N} \hat{u}_k \, T_k(x) \, . \tag{3.17}$$

The expansion coefficients $\hat{u}_k$, $k = 0, \ldots, N$, are determined by following the Galerkin-type technique described in Section 1.2.1. The residual $R_N = u - u_N$ is annuled in the weak average sense

$$(R_N, T_l)_w = 0 \, , \quad l = 0, \ldots, N \, , \tag{3.18}$$

namely,

$$\int_{-1}^{1} \left(u \, T_l \, w - \sum_{k=0}^{N} \hat{u}_k \, T_k \, T_l \, w \right) dx = 0 \, , \quad l = 0, \ldots, N \, .$$

Then, taking the orthogonality condition (3.12) into account, we obtain the expression for the Chebyshev expansion coefficients

$$\hat{u}_k = \frac{2}{\pi \, c_k} \int_{-1}^{1} u \, T_k \, w \, dx \, . \tag{3.19}$$

It seems worthwhile to express (3.17) by means of the representation (3.2), that is, $T_k = \cos kz$ with $x = \cos z$. The expansion (3.17) is then written as

$$u_N = \sum_{k=0}^{N} \hat{u}_k \, \cos kz \, , \tag{3.20}$$

showing that the Chebyshev expansion (3.17) with respect to x is equivalent to a cosine Fourier series in z. In fact, the function

$$v\,(z) = u\,(\cos z) = u\,(x)$$

defined in $0 \leq z \leq 2\pi$ is even and periodic since $v\,(z + 2\pi) = v\,(z)$. Moreover, $v\,(z)$ has as many bounded derivatives in $0 \leq z \leq \pi$ than $u\,(x)$ has in $-1 \leq x \leq 1$. Therefore, the convergence properties of the cosine Fourier series expansion (3.20) can be deduced from the results of Section 2.1.2. Moreover, since $v\,(z)$ is periodic, its representation (3.20) is continuous at the extremities $z = 0$ and $z = \pi$ and, consequently, is exempt from the Gibbs phenomenon at these points. More detailed results on the convergence of the Chebyshev approximation will be given in Section 3.6.

3.2.2 Differentiation

The expression of derivatives in the Chebyshev basis is more complicated than in the Fourier one. Indeed, the expression of the derivative of $T_k(x)$ involves all the polynomials of opposite parity and lower degree while the derivative of $e^{\underline{i}\,k\,x}$ is simply $\underline{i}\,k\,e^{\underline{i}\,k\,x}$. This makes the computational aspects of the two approximations very different : the Chebyshev differentiation matrices in the spectral and physical spaces are full while the analogous Fourier matrices are full only in the physical space.

From the recurrence relation (3.9), one obtains

$$T'_k(x) = 2\,k \sum_{n=0}^{K} \frac{1}{c_{k-1-2n}} T_{k-1-2n}(x)\,, \tag{3.21}$$

where $K = [(k-1)/2]$. Therefore, considering the first-order derivative

$$u'_N(x) = \sum_{k=0}^{N} \hat{u}_k\, T'_k(x) = \sum_{k=0}^{N} \hat{u}^{(1)}_k\, T_k(x) \tag{3.22}$$

and, taking Eq.(3.21) into account, we deduce the expression of the coefficient $\hat{u}^{(1)}_k$:

$$\hat{u}^{(1)}_k = \frac{2}{c_k} \sum_{\substack{p=k+1 \\ (p+k)\ \text{odd}}}^{N} p\,\hat{u}_p\,, \quad k = 0, \ldots, N-1\,, \tag{3.23}$$

and $\hat{u}^{(1)}_N = 0$. This can be written in matrix form as

$$\hat{U}^{(1)} = \hat{\mathcal{D}}\,\hat{U}\,, \tag{3.24}$$

where $\hat{U} = (\hat{u}_0, \ldots, \hat{u}_N)^T$, $\hat{U}^{(1)} = \left(\hat{u}^{(1)}_0, \ldots, \hat{u}^{(1)}_N\right)^T$ and $\hat{\mathcal{D}}$ is a strictly triangular upper matrix whose entries are deduced from (3.23).

The second-order derivative expansion is

$$u''_N(x) = \sum_{k=0}^{N} \hat{u}_k^{(2)}\, T_k(x) \tag{3.25}$$

with

$$\hat{u}_k^{(2)} = \frac{1}{c_k} \sum_{\substack{p=k+2 \\ (p+k)\ \text{even}}}^{N} p\, \left(p^2 - k^2\right)\, \hat{u}_p\,, \quad k = 0, \ldots, N-2 \tag{3.26}$$

and $\hat{u}_{N-1}^{(2)} = \hat{u}_N^{(2)} = 0$. This is written in matrix form as

$$\hat{U}^{(2)} = \hat{\mathcal{D}}^2\, \hat{U}\,, \tag{3.27}$$

where $\hat{U}^{(2)} = \left(\hat{u}_0^{(2)}, \ldots, \hat{u}_N^{(2)}\right)^T$.

The analytical expressions (3.23) and (3.25) are of interest each time the expansion coefficients of the derivatives are involved in algebraic calculations. On the other hand, if only the numerical values of the coefficients are needed, they can be calculated either from the matrix-vector products (3.24) and (3.27) or from recurrence formulas deduced from (3.9). More precisely, the expression for T_k given by (3.9) is brought into (3.21). Then, by identification of the derivative T'_k with the same index, we obtain the recurrence formula for the first-order derivative. The general recurrence formula for the coefficients $\hat{u}_k^{(p)}$ of the pth derivative is obtained by successive differentiations, let

$$c_{k-1}\, \hat{u}_{k-1}^{(p)} = \hat{u}_{k+1}^{(p)} + 2\, k\, \hat{u}_k^{(p-1)}\,, \quad k \geq 1\,, \tag{3.28}$$

be complemented with the starting values, for the first-order derivative

$$\hat{u}_N^{(1)} = 0\,, \quad \hat{u}_{N-1}^{(1)} = 2\, N\, \hat{u}_N\,, \tag{3.29}$$

and, for the second-order derivative,

$$\hat{u}_N^{(2)} = \hat{u}_{N-1}^{(2)} = 0\,, \quad \hat{u}_{N-2}^{(2)} = 2\,(N-1)\, \hat{u}_{N-1}^{(1)} = 2\, N\,(N-1)\, \hat{u}_N\,. \tag{3.30}$$

The recurrence relation (3.28) for $p = 2$ can be replaced by another connecting directly the coefficients $\hat{u}_k^{(2)}$ to $\hat{u}_k$. Such a relation is obtained by considering (3.28) with $p = 2$ and written for $k-1$ and $k+1$, such as the quantities $\hat{u}_{k-1}^{(1)}$ and $\hat{u}_{k+1}^{(1)}$ appear in the right-hand sides. Then these two equations are combined so that the quantities $\hat{u}_{k-1}^{(1)}$ and $\hat{u}_{k+1}^{(1)}$ can be eliminated thanks to (3.28) considered for $p = 1$. The resulting recurrence relation is

$$P_k\, \hat{u}_{k-2}^{(2)} + Q_k\, \hat{u}_k^{(2)} + R_k\, \hat{u}_{k+2}^{(2)} = \hat{u}_k\,, \quad 2 \leq k \leq N \tag{3.31}$$

with

$$P_k = \frac{c_{k-2}}{4\,k(k-1)}\,,\quad Q_k = \frac{-e_{k+2}}{2(k^2-1)}\,,\quad R_k = \frac{e_{k+4}}{4\,k(k+1)}\,, \tag{3.32}$$

where

$$e_j = \begin{cases} 1 & \text{if } j \le N\,, \\ 0 & \text{if } j > N\,. \end{cases} \tag{3.33}$$

Concerning the recurrent algorithm (3.28), Wengle and Seinfeld (1978) have remarked that it may be ill-conditioned in the sense that errors in the smallest coefficient $\hat{u}_k^{(p-1)}$ are amplified such that the accuracy of all the coefficients, even the largest ones $\hat{u}_k^{(p)}$, is destroyed. This can be avoided by simply equating to zero the coefficients smaller than a given threshold, depending on the accuracy of the computer.

3.3 Discrete Chebyshev series and collocation

This section is devoted to the Chebyshev collocation (i.e., interpolation) technique for the approximation of a given function. First, considering the discrete truncated Chebyshev series, the calculation of the expansion coefficients will be developed. Then the expression of the differentiation matrices will be established. Finally, we shall discuss an equivalent way to consider the Chebyshev expansion, namely by introducing the notion of Lagrange interpolation polynomial.

The collocation points considered here are the Gauss-Lobatto points defined by Eq.(3.8). Other sets of points, of similar nature (see Gottlieb *et al.*, 1984 ; Canuto *et al.*, 1988 ; Mercier, 1989), can be useful in some circumstances. For example, the choice of the set (3.7) may be of interest if it is not desired that the boundary points $x = \pm 1$ belong to the set of collocation points. Also, the Gauss-Radau points (see Appendix A) can be used if one wants to exclude the boundary point $x = -1$, for example in problems in cylindrical coordinates where $x = -1$ would correspond to the axis. On the other hand, for the solution of boundary value problems, the property of the set of collocation points held by the Gauss-Lobatto points to contain the boundaries is indispensable.

3.3.1 Calculation of Chebyshev coefficients

Considering the Chebyshev expansion (3.17), we want to calculate the coefficients $\hat{u}_k$ by means of the collocation (or interpolation) technique shown in Section 1.2.1. The technique consists of setting to zero the residual $R_N = u - u_N$ at the collocation points $x_i = \cos \pi\, i/N$, $i = 0, \ldots, N$,

let

$$u(x_i) = u_N(x_i) = \sum_{k=0}^{N} \hat{u}_k \, T_k(x_i) \,, \quad i = 0, \ldots, N \,. \tag{3.34}$$

By denoting $u_i = u(x_i) = u_N(x_i)$, and using the definition (3.1), the above equation gives :

$$u_i = \sum_{k=0}^{N} \hat{u}_k \, \cos \frac{k \, \pi \, i}{N} \,, \quad i = 0, \ldots, N \,. \tag{3.35}$$

Equation (3.34) [or (3.35)] gives an algebraic system of $2\,N+1$ equations for determining the $2\,N+1$ coefficients $\hat{u}_k$. The associated matrix $\mathcal{T} = [\cos k\,\pi\,i/N]$, $k, i = 0, \ldots, N$, is invertible ; as a matter of fact, it will be found below [Eq.(3.37)] that its inverse is $\mathcal{T}^{-1} = [2\,(\cos \pi\,i/N) \,/\, (\overline{c}_k \, \overline{c}_i \, N)]$, $k, i = 0, \ldots, N$.

The expression for the coefficients $\hat{u}_k$ (i.e., the solution of the system (3.34)) is directly obtained by means of the discrete orthogonality relation (3.16). By multiplying each side of (3.34) by $T_l(x_i)/\overline{c}_i$, then summing from $i = 0$ to $i = N$, and using the relation (3.16), we obtain

$$\hat{u}_k = \frac{2}{\overline{c}_k \, N} \sum_{i=0}^{N} \frac{1}{\overline{c}_i} u_i \, T_k(x_i) \,, \quad k = 0, \ldots, N \,, \tag{3.36}$$

or

$$\hat{u}_k = \frac{2}{\overline{c}_k \, N} \sum_{i=0}^{N} \frac{1}{\overline{c}_i} u_i \, \cos \frac{k \, \pi \, i}{N} \,, \quad k = 0, \ldots, N \,. \tag{3.37}$$

It must be noted that such expressions are nothing other than the numerical approximation (based on the Gauss-Lobatto points) of the integral appearing in Eq.(3.19).

The relations (3.35) and (3.36) show that the grid values u_i, as well as the coefficients $\hat{u}_k$, are related by truncated discrete Fourier series in cosine. Therefore, it is possible to use the FFT algorithm (in its cosine version) to connect the physical space (space of the grid values) to the spectral space (space of the coefficients). Note that it is also possible to simply make use of the matrix-vector products

$$U = \mathcal{T} \, \hat{U} \,, \quad \hat{U} = \mathcal{T}^{-1} \, U \,, \tag{3.38}$$

where U and $\hat{U}$ are, respectively, the vectors containing the grid values and the expansion coefficients. Note that the matrix-vector product for a moderate number of terms (namely 60-100) is less expensive in computing time than the FFT, depending on the computer and the routines used.

Results on the error of the collocation approximations (3.34) and (3.36) are given in Section 3.6.

3.3.2 Relation between collocation and Galerkin coefficients

The objective of this section is to make precise the relationship between the expansion coefficients defined by the integral (3.19) and those calculated from the sum (3.36). The first set of coefficients will be denoted here by $\hat{u}_k^e$ and the second set by $\hat{u}_k^c$. The general lines of the analysis made in Section 2.3 for the Fourier series apply in the present case. From Eqs.(3.19) and (3.36), we have

$$\hat{u}_k^e = \frac{2}{\pi\, c_k} \int_{-1}^{1} u(x)\, T_k(x)\, w(x)\, dx\,, \quad k = 0, \ldots, N\,, \tag{3.39}$$

$$\hat{u}_k^c = \frac{2}{\overline{c}_k\, N} \sum_{i=0}^{N} \frac{1}{\overline{c}_i} u(x_i)\, T_k(x_i)\,, \quad k = 0, \ldots, N\,. \tag{3.40}$$

Now we replace $u(x_i)$ in Eq.(3.40) by its expression in terms of the infinite series

$$u(x) = \sum_{k=0}^{\infty} \hat{u}_k^e\, T_k(x)$$

assumed to be absolutely convergent. Then, we decompose the resulting infinite sum into two partial sums according to

$$\begin{aligned} \hat{u}_k^c = \frac{2}{\overline{c}_k\, N} \sum_{l=0}^{N} \hat{u}_k^e \left[\sum_{i=0}^{N} \frac{1}{\overline{c}_i} T_k(x_i)\, T_l(x_i) \right] \\ + \frac{2}{\overline{c}_k\, N} \sum_{l=N+1}^{\infty} \hat{u}_k^e \left[\sum_{i=0}^{N} \frac{1}{\overline{c}_i} T_k(x_i)\, T_l(x_i) \right]\,. \end{aligned}$$

The expression appearing in square brackets is nothing other than the left-hand side of the discrete orthogonality relation (3.16). In the first bracket, the indices k and l vary between 0 and N, so that the relation (3.16) holds. On the other hand, in the second bracket, the index l varies between $N+1$ and infinity, so that the relation (3.16) is not applicable. The above expression for $\hat{u}_k^c$ can be written as

$$\hat{u}_k^c = \hat{u}_k^e + \frac{2}{\overline{c}_k\, N} \sum_{l=N+1}^{\infty} C_{k\,l}\, \hat{u}_l^e\,,$$

where

$$\begin{aligned} C_{k\,l} &= \sum_{i=0}^{N} \frac{1}{\overline{c}_i} T_k(x_i)\, T_l(x_i) = \sum_{i=0}^{N} \frac{1}{\overline{c}_i} \cos\frac{k\,i\,\pi}{N}\, \cos\frac{l\,i\,\pi}{N} \\ &= \frac{1}{2} \sum_{i=0}^{N} \frac{1}{\overline{c}_i} \left[\cos\frac{k-l}{N} i\,\pi + \cos\frac{k+l}{N} i\,\pi \right] \end{aligned}$$

with $k = 0, \ldots, N$ and $l = N+1, \ldots$. Then, by using the identity (valid for $p \in Z$),

$$\sum_{i=0}^{N} \cos \frac{p\, i\, \pi}{N} = \begin{cases} N+1 & \text{if } \; p = 2mN\,, \; m = 0, \pm 1, \pm 2, \ldots\,, \\ \dfrac{1}{2}\left[1 + (-1)^p\right] & \text{otherwise}\,, \end{cases}$$

we may calculate $C_{k\,l}$ and finally get the relation connecting the collocation coefficients to the Galerkin ones

$$\hat{u}_k^c = \hat{u}_k^e + \frac{1}{\overline{c}_k} \left[\sum_{\substack{m=1 \\ 2mN > N-k}}^{\infty} \hat{u}_{k+2mN}^e + \sum_{\substack{m=1 \\ 2mN > N+k}}^{\infty} \hat{u}_{-k+2mN}^e \right]. \quad (3.41)$$

The terms in square brackets are alias terms. The reason for their presence is the same as that for discrete Fourier series (Sections 2.2 and 2.3). This is a consequence of the fact that the Chebyshev expansion in x can also be considered as a cosine Fourier in the variable $z = \cos^{-1} x$.

3.3.3 Lagrange interpolation polynomial

Let us return to the approximation

$$u_N(x) = \sum_{k=0}^{N} \hat{u}_k\, T_k(x)\,, \quad (3.42)$$

where the coefficients $\hat{u}_k$, $k = 0, \ldots, N$, are determined by asking $u_N(x)$ to coincide with $u(x)$ at the collocation points $x_i = \cos \pi\, i/N$, $i = 0, \ldots, N$. Therefore, the polynomial of degree N defined by Eq.(3.42) is nothing other than the Lagrange interpolation polynomial based on the set $\{x_i\}$. Hence, it can also be written in the form

$$u_N(x) = \sum_{j=0}^{N} h_j(x)\, u(x_j) \quad (3.43)$$

with $u_N(x_j) = u(x_j)$, and $h_j(x)$ is the polynomial of degree N defined by

$$h_j\,(x) = \frac{(-1)^{j+1}\left(1 - x^2\right)\, T_N'(x)}{\overline{c}_j\, N^2\, (x - x_j)}\,. \quad (3.44)$$

This expression for h_j is easily constructed by recalling that the collocation points x_j are the zeros of the polynomial $\left(1 - x^2\right) T_N'(x)$ (see Section 3.1.1) and by observing that $\left(1 - x^2\right)\, T_N'\,(x)\,/\,(x - x_j) \;\to\; (-1)^{j+1}\, \overline{c}_j\, N^2$ when $x \to x_j$, $j = 0, \ldots, N$.

Therefore, the representation (3.43) is equivalent to (3.42) and is useful in several circumstances because it does not involve the spectral coefficients.

3.3.4 Differentiation in the physical space

As mentioned in Section 1.1.3, and discussed in Section 2.7.2 for the Fourier case, the application of collocation methods to the solution of differential equations may consider as unknowns the expansion coefficients as well as the grid values. The first approach is seldom employed in the Chebyshev case (see, e.g., Marion and Gay, 1986). On the other hand, the second approach (also known as "orthogonal collocation," see Finlayson, 1972) is of current use in computational fluid mechanics since the works by Wengle (1979) for the advection-diffusion equation and by Orszag and Patera (1983) or Ouazzani and Peyret (1984) for the Navier-Stokes equations.

Therefore, in the context of the collocation method where the unknowns are the grid values, it is necessary to express the derivatives at any collocation point in terms of the grid values of the function, that is, for the pth derivative $u_N^{(p)}$:

$$u_N^{(p)}(x_i) = \sum_{j=0}^{N} d_{i,j}^{(p)}\, u_N(x_j)\,, \quad i = 0, \ldots, N\,. \tag{3.45}$$

The coefficients $d_{i,j}^{(p)}$ can be calculated according to either of the following two ways :

(i) Eliminate $\hat{u}_k$ from the derivative

$$u_N^{(p)}(x_i) = \sum_{k=0}^{N} \hat{u}_k\, T_k^{(p)}(x_i)$$

by using expression (3.36). Then, express $T_k(x_i)$ and the pth derivative $T_k^{(p)}(x_i)$ in terms of trigonometrical functions according to $T_k = \cos kz$. Finally, apply the classical trigonometrical identities to evaluate the sums.

(ii) Differentiate p times directly the interpolation polynomial (3.43) :

$$u_N^{(p)}(x) = \sum_{j=0}^{N} h_j^{(p)}(x_i)\, u_N(x_j)\,.$$

Therefore, $d_{i,j}^{(p)} = h_j^{(p)}(x_i)$ which has to be evaluated from expression (3.44).

The expression of the coefficients $d_{i,j}^{(p)}$ for the first two derivatives are :

First-order derivative

$$\begin{aligned}
d_{i,j}^{(1)} &= \frac{\overline{c}_i}{\overline{c}_j}\,\frac{(-1)^{i+j}}{(x_i - x_j)}\,, && 0 \le i, j \le N\,,\ \ i \ne j \\
d_{i,i}^{(1)} &= -\frac{x_i}{2(1 - x_i^2)}\,, && 1 \le i \le N-1 \\
d_{0,0}^{(1)} &= -d_{N,N}^{(1)} = \frac{2\,N^2 + 1}{6}\,,
\end{aligned} \tag{3.46}$$

where $x_i = \cos(\pi i/N)$, $\overline{c}_0 = \overline{c}_N = 2$, $\overline{c}_j = 1$ for $1 \le j \le N-1$.

Second-order derivative

$$
\begin{aligned}
d^{(2)}_{i,j} &= \frac{(-1)^{i+j}}{\overline{c}_j}\,\frac{x_i^2 + x_i\,x_j - 2}{(1-x_i^2)(x_i - x_j)^2}\,, && 1 \le i \le N-1\,,\\
&&& 0 \le j \le N\,,\ \ i \ne j\\
d^{(2)}_{i,i} &= -\frac{(N^2-1)(1-x_i^2)+3}{3(1-x_i^2)^2}\,, && 1 \le i \le N-1\\
d^{(2)}_{0,j} &= \frac{2}{3}\,\frac{(-1)^j}{\overline{c}_j}\,\frac{(2\,N^2+1)(1-x_j)-6}{(1-x_j)^2}\,, && 1 \le j \le N\\
d^{(2)}_{N,j} &= \frac{2}{3}\,\frac{(-1)^{j+N}}{\overline{c}_j}\,\frac{(2\,N^2+1)(1+x_j)-6}{(1+x_j)^2}\,, && 0 \le j \le N-1\\
d^{(2)}_{0,0} &= d^{(2)}_{N,N} = \frac{N^4-1}{15}\,.
\end{aligned}
\tag{3.47}
$$

It may be useful to recall that

$$
d^{(2)}_{i,j} = \sum_{k=0}^{N} d^{(1)}_{i,k}\,d^{(1)}_{k,i}\,. \tag{3.48}
$$

In vector form, the derivatives may be expressed as

$$
U^{(1)} = \mathcal{D}\,U\,, \quad U^{(2)} = \mathcal{D}^2 U \tag{3.49}
$$

where

$$
U = \left(u_N(x_0), \ldots, u_N(x_N)\right)^T\,, \quad U^{(p)} = \left(u_N^{(p)}(x_0), \ldots, u_N^{(p)}(x_N)\right)^T \tag{3.50}
$$

with $p = 1, 2$. The differentiation matrix $\mathcal{D}$ is defined by

$$
\mathcal{D} = \left[d^{(1)}_{i,j}\right]\,, \quad i,j = 0, \ldots, N\,. \tag{3.51}
$$

3.3.5 Round-off errors

An important question, when dealing with Chebyshev approximations of high degree N, concerns the effect of round-off errors. A possible cause of error associated with the use of the recurrence formulas for calculating the coefficients of the Chebyshev expansion of derivatives, has been mentioned in Section 3.2.2. Another way to calculate the derivatives is to make use of the differentiation matrices whose entries have been given in the previous section. This technique is very often employed because it is not necessary

to have recourse to the FFT algorithm, although this latter is much more economical for high resolution.

The possible sources of the magnification of the round-off errors, associated with the calculation of derivatives through matrix-vector products, have been analyzed in several works (Rothman, 1991 ; Breuer and Everson, 1992 ; Bayliss *et al.*, 1994).

One cause is the disparity in magnitude between the entries of the matrices. For example, in the case of the first-order differentiation matrix $\mathcal{D} = [d_{i,j}^{(1)}]$, the smallest elements are $O(1)$ while the largest ones are $O(N^2)$. Another cause can be found in the computation of the matrix elements which involve the subtraction of nearly equal numbers, as observed by Rothman (1991) who proposes a simple way to minimize these round-off errors. The technique consists of using trigonometrical identities to express the quantity $(x_i - x_j)$ in $d_{i,j}^{(1)}$ and $(1 - x_i)^2$ in $d_{i,i}^{(1)}$, let

$$x_i - x_j = 2 \sin \frac{(j+i)\,\pi}{2\,N} \sin \frac{(j-i)\,\pi}{2\,N}\,, \quad 1 - x_i^2 = \sin^2 \frac{i\,\pi}{N}\,. \tag{3.52}$$

Numerical experiments have shown the efficiency of this simple procedure. On the other hand, it is inefficient and even worse when applied to the second-order differentiation matrix $\mathcal{D}^2$. For such a matrix, Rothman recommends not using the analytical expressions $d_{i,j}^{(2)}$ given in the previous section but rather to calculate numerically $\mathcal{D}^2$ as the square of the matrix $\mathcal{D}$ whose entries $d_{i,j}^{(1)}$ are computed using the trigonometrical relations (3.52).

Another cause of round-off errors, identified by Bayliss *et al.* (1994), is the failure of the computed differentiation matrix to represent exactly the derivative of a constant, that is, to numerically satisfy the identity

$$\sum_{j=0}^{N} d_{i,j}^{(1)} = 0\,, \quad i = 0, \ldots, N\,. \tag{3.53}$$

The remedy suggested by Bayliss *et al.* (1994) is to calculate the off-diagonal entries $d_{i,j}^{(1)}$, $j \neq i$, by means of the formulas (3.46), and the diagonal entries $d_{i,i}^{(1)}$ by

$$d_{i,i}^{(1)} = -\sum_{\substack{j=0 \\ j \neq i}}^{N} d_{i,j}^{(1)}\,, \quad i = 0, \ldots, N\,. \tag{3.54}$$

Such a procedure gives a substantial improvement concerning the effect of round-off errors. Compared to the above technique consisting of the use of trigonometrical expressions, the present technique (Bayliss *et al.*, 1994) gives much better results. However, the use of the trigonometrical

relations (3.52) in the correction technique of Bayliss *et al.* (1994) does not bring any improvement as shown by numerical experiments performed on various functions.

In conclusion, for the calculation of the first-order derivative, we recommend using the correction technique of Bayliss *et al.* (1994) with the entries $d_{i,j}^{(1)}$ (for $j \neq i$) calculated with the analytic expressions given in (3.46) (without the use of trigonometrical expressions) and the diagonal entries $d_{i,i}^{(1)}$ with Eq.(3.54).

For the calculation of the second-order derivative, a large number of possibilities can be envisaged by combining the various techniques invoked above. From the numerical experiments performed, any technique has been found better than the other ones for all the tested functions. However, the technique based on the Bayliss *et al.* correction often gives the best results and, when this is not the case, the results are not far from the best ones.

In conclusion, for the calculation of the second-order derivative we recommend the following technique :

1. Calculate a provisional second-order differentiation matrix $\tilde{\mathcal{D}}^2$ as the square product of the matrix $\mathcal{D}$ whose entries $d_{i,j}^{(1)}$ are calculated following the correction technique of Bayliss *et al.*, based on Eq.(3.46) and (3.54).

2. From the entries $\tilde{d}_{i,j}^{(2)}$ of $\tilde{\mathcal{D}}^2$, calculate the entries $d_{i,j}^{(2)}$ of the final second-order matrix $\mathcal{D}^2$ by a repeated application of the Bayliss *et al.* correction technique, let

$$\begin{aligned} d_{i,j}^{(2)} &= \tilde{d}_{i,j}^{(2)} \quad \text{for} \quad j \neq i \,, \\ d_{i,i}^{(2)} &= - \sum_{\substack{j=0 \\ j \neq i}}^{N} \tilde{d}_{i,j}^{(2)} \,, \quad i = 0, \ldots, N \,. \end{aligned} \tag{3.55}$$

It is important to note that, in a general way, the effect of round-off errors on the calculation of the derivatives is significant when the polynomial degree N is larger than the value N_0 needed to represent the function itself within the machine accuracy. In other words, the value of N, for which the computation of the derivatives is strongly subject to the amplification of round-off errors, depends on the function under consideration, so that, in some cases, it can be rather large. In conclusion, it is recommended to avoid, as much as possible, over-resolution.

Moreover, it has been observed that the errors on the second-order (or higher) derivative are always larger than those associated with the first-order derivative.

The solution of differential equations makes use of the differentiation matrices to construct the approximate discrete operators. As pointed out by Breuer and Everson (1992), and generally observed in the calculations (see Section 3.4.3), the effect of round-off errors is much less marked when

solving differential equations than when computing the derivatives of a function.

3.3.6 Relationship with finite-difference and similar approximations

Formula (3.45) makes clear the relationship between finite-difference and collocation Chebyshev approximations. In a finite-difference method, the approximation of a derivative at a grid point involves only very few neighbouring grid values of the function, while the Chebyshev approximation involves all the grid values. The finite-difference formula approximating a derivative can be formally obtained by representing the function under consideration through a local Lagrange interpolation polynomial of low degree (Ferziger, 1981), this polynomial changing from one discretization point to another. In the Chebyshev method, the interpolation polynomial is the same in the whole domain, it involves the values of the function at all collocation points and, consequently, the formula expressing the derivative, like (3.45), also involves all the grid values.

Higher-order finite-difference-type approximations can be constructed through Hermitian methods, which amount to considering a local Hermite interpolation polynomial (Peyret, 2000), and are closely connected to spline-function approximations (Rubin and Khosla, 1977). These approximations, like the classical finite-difference methods, the finite-volume methods and the finite-element methods are local, contrary to the Chebyshev (as well as Fourier) approximation which is of a global nature.

The global character of spectral methods is beneficial for accuracy. On the other hand, the solution of a differential problem at a given point is strongly dependent on the solution at all other points and, in particular, at the boundaries. As a consequence, the influence of the way in which the boundary conditions are handled is not localized, as is often the case with local-type methods, but extends to the whole computational domain. Hence, great care must be taken when prescribing the boundary conditions. For the same reason, the presence of local singularities may contaminate the solution everywhere even at large distances through Gibbs-type oscillations. Therefore, it is recommended employing spectral methods for computing sufficiently smooth functions. However, sophisticated filtering methods (Gottlieb and Shu, 1997), domain decomposition methods, and techniques of subtraction of the singularity constitute interesting approaches to the solution of singular problems. The latter two approaches are discussed in Chapter 8.

Moreover, the matrices associated with the Chebyshev approximation of a differential problem are full, contrary to the local-type approximations. These matrices are not symmetric nor skew-symmetric, and they are generally ill-conditioned. Therefore, great care must also be taken when

solving the corresponding algebraic systems. This point will be discussed in Section 3.4.

In spite of these apparent drawbacks ("apparent" because remedies exist for curing them), use of the spectral method is recommended for the representation of smooth solutions when high accuracy is required. The error associated with the Chebyshev (as well as Fourier) approximation is $O(1/N^m)$ where N refers to the truncation and m is connected to the number of continuous derivatives of the function under consideration (see Section 3.6 for Chebyshev methods and Section 2.1.2 for Fourier methods). In particular, for infinitely differentiable functions m is larger than any integer and then exponential accuracy is obtained. Such behaviour has to be compared to the $O(1/N^p)$ error of a local-type approximation, like the finite-difference method, where $1/N$ is the mesh size and p, which depends on the method, is essentially finite and even generally small.

Another advantage of the spectral approximation is that it is defined everywhere in the computational domain. Therefore, it is easy to get an accurate value of the function under consideration at any point of the domain, beside the collocation points. This property is often exploited, in particular to get a significant graphic representation of the solution, making apparent the possible oscillations due to a wrong approximation of the derivative.

Finally, an additional property of the spectral methods is the easiness with which the accuracy of the computed solution can be estimated. This can be done by simply checking the decrease of the spectral coefficients. There is no need to perform several calculations by modifying the resolution, as is usually done in finite-difference and similar methods for estimating the "grid-convergence".

3.4 Differential equation with constant coefficients

The application of the Chebyshev method to the solution of a one-dimensional boundary-value problem with constant coefficients is the objective of the present section. The tau method and the collocation method are successively described. The properties of each of these methods are discussed and compared by considering the numerical solution of the one-dimensional Helmholtz equation.

Let us consider the second-order differential problem

$$-\nu\, u'' + a\, u' + b\, u = f\,, \quad -1 < x < 1 \tag{3.56}$$

$$\alpha_-\, u(-1) + \beta_-\, u'(-1) = g_-\,, \tag{3.57}$$

$$\alpha_+\, u(1) + \beta_+\, u'(1) = g_+\,, \tag{3.58}$$

where f is a given function of x ; ν, a, b, $\alpha_\pm$, $\beta_\pm$, and $g_\pm$ are constant. In Section 1.3.1, we have introduced the traditional Galerkin method which

applies when the basis functions $\varphi_k(x)$ satisfy the homogeneous boundary conditions, namely, Eqs.(3.57) and (3.58) with $g_- = g_+ = 0$. The Chebyshev polynomials do not meet this requirement. However, in some situations, it is possible to construct a basis verifying the homogeneous boundary conditions. For example, in the case of the Dirichlet conditions $u(\pm 1) = 0$, the suitable basis functions are

$$\varphi_k(x) = \begin{cases} T_k(x) - T_0 = T_k(x) - 1 & \text{if } k \text{ is even}, \\ T_k(x) - T_1(x) & \text{if } k \text{ is odd}. \end{cases}$$

However, the basis $\{\varphi_k\}$ is not orthogonal and such a Galerkin method is seldomly used. It is generally replaced by the simpler method known under the name of tau method, although this latter itself is now superseded by the more versatile collocation method.

3.4.1 Tau method

The solution $u(x)$ of Eqs.(3.56)-(3.58) is approximated with $u_N(x)$ defined by

$$u_N(x) = \sum_{k=0}^{N} \hat{u}_k \, T_k(x) \tag{3.59}$$

and the residual associated to the differential equation (3.56) is

$$R_N = -\nu \, u_N'' + a \, u_N' + b \, u_N - f \, . \tag{3.60}$$

The tau equations are obtained by setting to zero the first $N-2$ scalar products

$$(R_N, T_i)_w = 0 \, , \quad i = 0, \ldots, N-2 \, , \tag{3.61}$$

and by adding the boundary conditions (3.57) and (3.58). The derivatives u_N' and u_N'' are expressed respectively by Eqs.(3.22) and (3.25) so that Eq.(3.61) gives

$$(R_N, T_i)_w = \sum_{k=0}^{N} \Big(-\nu \, \hat{u}_k^{(2)} + a \, \hat{u}_k^{(1)} + b \, \hat{u}_k \Big) \int_{-1}^{1} T_k \, T_i \, w \, dx - \int_{-1}^{1} f \, T_i \, w \, dx = 0$$

for $i = 0, \ldots, N-2$. Then, thanks to the orthogonality relation (3.12), we get

$$-\nu \, \hat{u}_k^{(2)} + a \, \hat{u}_k^{(1)} + b \, \hat{u}_k = \hat{f}_k \, , \quad k = 0, \ldots, N-2 \, , \tag{3.62}$$

where $\hat{f}_k = \int_{-1}^{1} f \, T_k \, w \, dx$ is the coefficient of the Chebyshev expansion of f. These equations are supplemented with the boundary conditions (3.57) and (3.58) which yield, taking (3.5) into account,

$$\sum_{k=0}^{N} (-1)^k \left(\alpha_- - \beta_- \, k^2 \right) \, \hat{u}_k = g_- \tag{3.63}$$

$$\sum_{k=0}^{N} \left(\alpha_+ + \beta_+ k^2\right) \hat{u}_k = g_+ \,. \tag{3.64}$$

Finally, by replacing $\hat{u}_k^{(1)}$ and $\hat{u}_k^{(2)}$ in Eq.(3.62) by their respective expressions (3.23) and (3.26) in terms of the coefficients $\hat{u}_k$, $k = 0, \ldots, N$, the system (3.62)-(3.64) can be written in the form

$$\mathcal{A}\,\hat{U} = F \,, \tag{3.65}$$

where $\hat{U} = (\hat{u}_0, \ldots, \hat{u}_N)^T$, $F = \left(\hat{f}_0, \ldots, \hat{f}_{N-2}, g_-, g_+\right)^T$, and $\mathcal{A} = [a_{ij}]$, $i, j = 0, \ldots, N$, is the $(N+1) \times (N+1)$ matrix constructed from the matrix

$$\mathcal{Q} = -\nu\,\hat{\mathcal{D}}^2 + a\,\hat{\mathcal{D}} + b\,\mathcal{I} \,, \tag{3.66}$$

[where $\hat{\mathcal{D}}$ is defined by Eqs.(3.23), (3.24), and $\mathcal{I}$ is the identity matrix] in which the two last lines are, respectively, replaced by

$$\begin{aligned} a_{N-1,j} &= (-1)^j \left(\alpha_- - \beta_- j^2\right) , \\ a_{N,j} &= \alpha_+ + \beta_+ j^2 \,, \qquad j = 0, \ldots, N. \end{aligned}$$

The matrix $\mathcal{A}$ is ill-conditioned so that the solution of the system (3.65) requires the use of adapted methods. However, we do not go further in that direction for two reasons. First, the tau method is less and less employed in favour of the collocation method. Second, when considering the Navier-Stokes equations (which is the main subject of the present book) the equations to be solved are of Helmholtz type [i.e., Eq.(3.56) with $a = 0$] which can be solved very efficiently as will be discussed below. The absence of the first-order derivative comes from the fact that the first-order derivative terms in the Navier-Stokes equations, being nonlinear, are generally handled in an explicit way and, hence, are not involved in the algebraic system to be solved.

In the case of the Helmholtz equation with Dirichlet or Neumann boundary conditions, the solution of the resulting system (3.65) reduces to the solution of two quasi-tridiagonal systems (Elliott, 1961 ; Haidvogel, 1977; Gottlieb and Orszag, 1977) whose construction is now described in the Dirichlet case.

The Dirichlet problem for the Helmholtz equation is

$$-\nu\,u'' + b\,u = f \,, \quad -1 < x < 1 \tag{3.67}$$

$$u(-1) = g_- \,, \quad u(1) = g_+ \,, \tag{3.68}$$

and the tau equations, deduced from (3.62) and (3.64), are

$$-\nu\,\hat{u}_k^{(2)} + b\,\hat{u}_k = \hat{f}_k \,, \quad k = 0, \ldots, N-2 \,, \tag{3.69}$$

$$\sum_{k=0}^{N}(-1)^k \hat{u}_k = g_- \,, \tag{3.70}$$

$$\sum_{k=0}^{N} \hat{u}_k = g_+ \,. \tag{3.71}$$

Now the goal is to eliminate the coefficients $\hat{u}_k^{(2)}$ thanks to the recurrence relation (3.31). More precisely, denoting Eq.(3.69) by E_k, we construct the linear combination $P_k\, E_{k-2} + Q_k\, E_k + R_k\, E_{k+2}$, with P_k, Q_k, and R_k as defined in (3.32). Then, using Eq.(3.31), we get the equations

$$P'_k\, \hat{u}_{k-2} + Q'_k\, \hat{u}_k + R'_k\, \hat{u}_{k+2} = \varphi_k \,, \quad k = 2, \ldots, N \,, \tag{3.72}$$

where

$$P'_k = b\, P_k \,, \quad Q'_k = b\, Q_k - \nu \,, \quad R'_k = b\, R_k \,,$$

and

$$\varphi_k = P_k\, \hat{f}_{k-2} + Q_k\, \hat{f}_k + R_k\, \hat{f}_{k+2} \,.$$

It is easy to see that Eqs (3.72) lead to two uncoupled systems : one for the even coefficients and the other for the odd ones. These systems are closed by adding supplementary equations obtained from the boundary conditions. By addition of Eqs (3.70) and (3.71) we get an equation involving the even coefficients only, and by subtraction a similar equation is obtained for the odd modes, that is

$$\hat{u}_0 + \hat{u}_2 + \ldots = \frac{1}{2}\,(g_+ + g_-) \,, \tag{3.73}$$

$$\hat{u}_1 + \hat{u}_3 + \ldots = \frac{1}{2}\,(g_+ - g_-) \,. \tag{3.74}$$

Therefore, the sets of even and odd coefficients are, respectively, the solutions of uncoupled algebraic systems. The associated matrices have a quasi-tridiagonal structure (three nonzero diagonals and one row) which allows a direct solution to be obtained from an extension of the usual $L\,U$ decomposition algorithm. The reader will find in Appendix B the description of the algorithm as developed by Thual (1986). It is interesting to note that the operation count of the algorithm is $O(N)$ as for the pure tridiagonal system. Also, the algorithm is stable, that is, the solution is bounded, if the considered matrices satisfy the diagonally dominance-type conditions (see Appendix B). For $\nu > 0$, $b \geq 0$ (the usual case in the application to unsteady problems where b is essentially the reciprocal of the time-step), these conditions are satisfied. On the other hand, when $b < 0$, the diagonally dominant conditions are satisfied if $\nu > -b/3$.

The method described above for the Dirichlet problem applies to the case of the Neumann conditions but not to the general Robin-type boundary

conditions (3.57)-(3.58). In the case where the first-order derivative term is present in the differential equation, the application of the method leads to a quasi-pentadiagonal algebraic system which can be solved with an analogous algorithm (Dennis and Quartapelle, 1985).

3.4.2 Collocation method

The fundamentals of the collocation method have been given in Section 1.3.2. It has been observed that the unknowns can be either the coefficients $\hat{u}_k$, $k = 0, \ldots, N$, of the expansion or the values $u_N(x_i)$, $i = 0, \ldots, N$, of the approximate solution at the collocation points x_i. Although both techniques are equivalent from the mathematical point of view, the discrete equations, however, are obviously different. The first technique has received little interest in the field of fluid mechanics. We refer to Marion and Gay (1986) for an application to the solution of the Navier-Stokes equations. On the other hand, the second technique, consisting of considering the grid values $u_N(x_i)$ as unknowns, is of current application. As a matter of fact, as mentioned in Section 3.3.3, this technique amounts to approximating the solution $u(x)$ with a polynomial $u_N(x)$ of degree at most equal to N, namely, $u_N \in \mathbb{P}_N$. Then, the differential equation is forced to be satisfied exactly by this polynomial at the inner collocation points.

Let us consider the differential problem (3.56)-(3.58) and the Gauss-Lobatto collocation points :

$$x_i = \cos \frac{\pi\, i}{N}\,, \quad i = 0, \ldots, N\,. \tag{3.75}$$

By setting to zero the residual $R_N(x)$ [defined by Eq.(3.60)] at the inner collocation points x_i, $i = 1, \ldots, N-1$, and by adding the boundary conditions, we obtain the collocation equations

$$\begin{aligned}
&-\nu\, u_N''(x_i) + a\, u_N'(x_i) + b\, u_N(x_i) = f(x_i)\,, \quad i = 1, \ldots, N-1 \\
&\alpha_-\, u_N(x_N) + \beta_-\, u_N'(x_N) = g_- \\
&\alpha_+\, u_N(x_0) + \beta_+\, u_N'(x_0) = g_+\,.
\end{aligned} \tag{3.76}$$

Now, from (3.45), these equations give

$$\begin{aligned}
&\sum_{j=0}^{N} \left(-\nu\, d_{i,j}^{(2)} + a\, d_{i,j}^{(1)}\right) u_N(x_j) + b\, u_N(x_i) = f(x_i)\,, \quad i = 1, \ldots, N-1 \\
&\alpha_-\, u_N(x_N) + \beta_- \sum_{j=0}^{N} d_{N,j}^{(1)}\, u_N(x_j) = g_- \\
&\alpha_+\, u_N(x_0) + \beta_+ \sum_{j=0}^{N} d_{0,j}^{(1)}\, u_N(x_j) = g_+\,,
\end{aligned} \tag{3.77}$$

or, in matrix form,

$$\mathcal{A}\,U = F\,, \tag{3.78}$$

where $U = (u_N(x_0), \ldots, u_N(x_N))^T$, $F = (g_+, f_1, \ldots, f_{N-1}, g_-)^T$, and $\mathcal{A}$ is the $(N+1) \times (N+1)$ matrix constructed from the matrix

$$\mathcal{Q} = -\nu\,\mathcal{D}^2 + a\,\mathcal{D} + b\,\mathcal{I}$$

[$\mathcal{D}$ is the differentiation matrix defined by Eq.(3.51) and $\mathcal{I}$ is the identity matrix] in which the first and last lines are replaced by quantities coming from the boundary conditions. More precisely, the entries $a_{i,j}$ of the matrix $\mathcal{A}$ have the following expression

$$a_{0,j} = \alpha_+ + \beta_+ \sum_{j=0}^{N} d_{0,j}^{(1)}\,, \quad a_{N,j} = \alpha_- + \beta_- \sum_{j=0}^{N} d_{N,j}^{(1)}\,, \quad j = 0, \ldots, N\,,$$

$$a_{i,j} = -\nu\,d_{i,j}^{(2)} + a\,d_{i,j}^{(1)} + b\,\delta_{i,j}\,, \quad i = 1, \ldots, N-1\,, \quad j = 0, \ldots, N\,.$$

Various methods are available for the solution of the system (3.78). However, it must be noticed that problems like (3.56)-(3.58) have generally to be solved at each time-cycle of an unsteady process. That is, they have to be solved a very large number of times, so the solution method must take this peculiarity into account. Therefore, an efficient method is the one which necessitates the least calculations possible at each time-cycle, leaving in a preprocessing stage, performed before the start of the time-integration, the calculations which can be done only once. Moreover, it is important that the calculations done at each time-cycle be adapted to modern computers taking advantage of vectorization and parallelization. In these respects, the method based on the $L\,U$ decomposition, which constitutes an accurate way to solve the system (3.78), is not necessarily the most economical from the point of view of vectorization.

Moreover, the properties of the matrix $\mathcal{A}$ have also to be taken into account. The matrix $\mathcal{A}$ has no good properties : it is full, not symmetric nor skew-symmetric, and it is ill-conditioned. To be more precise, Tables 3.1 and 3.2 give information on the behaviour of the spectral radius ρ and the condition number κ of the operators associated with the first- and second-order derivatives supplied with various types of boundary conditions. We recall that, for an $n \times n$ matrix $\mathcal{M}$ with eigenvalues $\lambda_i\,(\mathcal{M})$, $i = 1, \ldots, n$, such that $\lambda_{max}\,(\mathcal{M}) = \max_i |\lambda_i\,(\mathcal{M})|$ and $\lambda_{min}\,(\mathcal{M}) = \min_i |\lambda_i\,(\mathcal{M})|$, the spectral radius $\rho\,(\mathcal{M})$ is defined by

$$\rho\,(\mathcal{M}) = \lambda_{max}\,(\mathcal{M}) \tag{3.79}$$

and the condition number $\kappa\,(\mathcal{M})$ is defined by

$$\kappa\,(\mathcal{M}) = \left[\frac{\lambda_{max}\left(\mathcal{M}^T\mathcal{M}\right)}{\lambda_{min}\left(\mathcal{M}^T\mathcal{M}\right)}\right]^{1/2}\,. \tag{3.80}$$

Operator	Boundary conditions	ρ	κ
u'	$u(-1) = 0$	$0.089N^2$ (40)	$0.049N^2$ (40)
u''	$u(-1) = 0$ $u(1) = 0$	$0.047N^4$ (50)	$0.020N^4$ (50)
u''	$u(-1) = 0$ $u'(1) = 0$	$0.047N^4$ (50)	$0.086N^4$ (50)
u''	$u'(-1) = 0$ $u'(1) = 0$	$0.014N^4$ (60)	$(\star)$ $0.0065N^4$ (30)

Table 3.1. Spectral radius ρ and condition number κ of the Chebyshev matrices approximating first- and second-order derivatives with various boundary conditions (the boundary values have been eliminated). The numbers in parentheses refer to values of N for which the given figures are correct. In $(\star)$ the null eigenvalue has been discarded.

Table 3.1 displays the behaviour, with respect to N of the spectral radius ρ and the condition number κ of the discrete operators, in the case where the boundary values have been eliminated thanks to the boundary conditions. Table 3.2 gives the same information for the full $(N+1) \times (N+1)$ matrices without elimination of the boundary values.

A good way to deal with the system (3.78) is to invert $\mathcal{A}$ in the preprocessing stage and to store its inverse $\mathcal{A}^{-1}$. So that, at each time-cycle, only one matrix-vector product has to be performed, this being efficiently done on vector computers. However, taking the ill-conditioning of $\mathcal{A}$ into account, the inversion has to be done by a routine whose accuracy is very reliable, for example, LINRG from IMSL library or F04AEF from NAG. Both routines are based on the LU decomposition. The first one gives directly the inverse $\mathcal{A}^{-1}$. The second one solves a set of equations

$$\mathcal{A}\,U^k = F^k$$

with $F^k = [\delta_{k,j}]$, $j = 0, \ldots, N$, then $\mathcal{A}^{-1}$ is the matrix constructed such that each column is made with the elements of one vector U^k.

Besides the use of an efficient inversion routine, it is recommended applying a suitable preconditionning before the inversion. A good preconditioner is the finite-difference centered operator associated with the discretization of the problem (3.56), (3.57) based on the Gauss-Lobatto points (3.75). Let $\mathcal{A}_0$ be the resulting tridiagonal matrix and let $\mathcal{A}_0^{-1}$ be its inverse, Eq.(3.78) is multiplied at the left by $\mathcal{A}_0^{-1}$, i.e.,

$$\mathcal{A}_0^{-1}\,\mathcal{A}\,U = \mathcal{A}_0^{-1}\,F\,.$$

Operator	Boundary conditions	ρ	κ
u'	$u(-1)=0$	$0.089N^2$ (40)	$0.56N^{5/2}$ (50)
u''	$u(-1)=0$ $u(1)=0$	$0.047N^4$ (50)	$0.041N^{9/2}$ (30)
u''	$u(-1)=0$ $u'(1)=0$	$0.047N^4$ (50)	$0.088N^{9/2}$ (60)
u''	$u'(-1)=0$ $u'(1)=0$	$0.047N^4$ (80)	$(\star)\ 0.029N^{9/2}$ (60)

Table 3.2. Spectral radius ρ and condition number κ of the Chebyshev matrices approximating first- and second-order derivatives with various boundary conditions (the boundary values are not eliminated). The numbers in parentheses refer to values of N for which the given figures are correct. In $(\star)$ the null eigenvalue has been discarded.

The matrix $\mathcal{B} = \mathcal{A}_0^{-1}\,\mathcal{A}$ is found to be well-conditioned so that it can be inverted with negligible round-off errors. The solution of (3.73) is then calculated as $U = \mathcal{B}^{-1}\,\mathcal{A}_0^{-1}\,F$.

We refer to Section 3.6 for some results concerning the approximation error associated with the collocation and tau methods for solving the one-dimensional Poisson equation with Dirichlet conditions.

3.4.3 Error equation

The polynomial $u_N(x)$, obtained from the application of the tau method (Section 3.4.1) or the collocation method (Section 3.4.2), is an approximate solution of the problem (3.56)-(3.58), that is

$$L\,u = f\,,\quad -1 < x < 1 \tag{3.81}$$

$$B_-u(-1) = g_-\,,\quad B_+u(1) = g_+\,. \tag{3.82}$$

It may be interesting to know what the equation is that is exactly solved by $u_N(x)$ (with the same boundary conditions). This equation, called the "error equation" (Canuto *et al.*, 1988), has been considered by Gottlieb and Orszag (1977) and, subsequently, by several authors for theoretical studies on convergence and the stability of various spectral approximations. In Chapter 7, the error equation will be used to analyze the error on the divergence of the velocity in the solution of the Navier-Stokes equations when using the influence matrix technique. The construction of the error equation is described now by successively considering the tau and the collocation methods.

(a) Tau method

For the problem (3.81)-(3.82) the error equation is constructed from the equation

$$L\,u = f + p\,, \tag{3.83}$$

where p is a function of x which is assumed to be represented in the Chebyshev basis by

$$p(x) = \sum_{k=0}^{\infty} \tau_k\, T_k(x)\,. \tag{3.84}$$

The coefficients τ_k are determined such that the polynomial u_N, the approximate solution of Eqs.(3.82)-(3.83) according to the tau method, satisfies the same problem but for all values of k, so that u_N will be the exact solution. The Galerkin equations associated with Eq.(3.83) are

$$(L\,u_N - f - p\,,\, T_i)_w = 0\,,\quad i = 0, \ldots, \infty\,. \tag{3.85}$$

For $i = 0, \ldots, N-2$, these equations give

$$\frac{\pi}{2}\, c_i\, \tau_i = (L\,u_N - f_N\,,\, T_i)_w\,,$$

where $f_N = \sum_{k=0}^{N} \hat{f}_k\, T_k$ is the Chebyshev approximation to $f(x)$. Then, Eq.(3.85) yields $\tau_i = 0$, since u_N satisfies the tau equations

$$(L\,u_N - f_N\,,\, T_i)_w = 0\,,\quad i = 0, \ldots, N-2\,.$$

Then, for $i = N-1, N$, Eq.(3.85) yields

$$\frac{\pi}{2}\tau_{N-1} = (L\,u_N - f_N\,,\, T_{N-1})_w = (L\,u_N\,,\, T_{N-1})_w - \frac{\pi}{2}\hat{f}_{N-1}\,,$$

$$\frac{\pi}{2}\tau_N = (L\,u_N - f_N\,,\, T_N)_w = (L\,u_N\,,\, T_N)_w - \frac{\pi}{2}\hat{f}_N\,.$$

Finally, for $i = N+1, \ldots, \infty$, Eq.(3.85) simply gives

$$\tau_i = -\hat{f}_i\,.$$

By collecting these results, we obtain, from Eq.(3.83), the error equation

$$L\,u = f_{N-2} + \tau'_{N-1}\, T_{N-1} + \tau'_N\, T_N\,, \tag{3.86}$$

where f_{N-2} is the $N-2$ truncated Chebyshev expansion of f and

$$\tau'_{N-1} = \frac{2}{\pi}\,(L\,u_N\,,\, T_{N-1})_w\,,\quad \tau'_N = \frac{2}{\pi}\,(L\,u_N\,,\, T_N)_w\,. \tag{3.87}$$

It may be observed that the error term $\tau_{N-1}T_{N-1} + \tau_N T_N$ is coming from the Galerkin equations corresponding to the last two test functions which have been discarded and replaced by the boundary conditions.

Obviously, the right-hand side of the error equation (3.86) is a polynomial of degree at most N, and the exact solution of this equation that satisfies the boundary conditions (3.82) is the polynomial u_N, the solution of the tau equations of Section 3.4.1.

For the differential operator L defined by Eq.(3.56) the error equation is

$$L\,u = f_{N-2} + (2\,a\,N\,\hat{u}_N + b\,\hat{u}_{N-1})\,T_{N-1} + b\,\hat{u}_N\,T_N,$$

where the relation $\hat{u}^{(1)}_{N-1} = 2\,N\,\hat{u}_N$ has been taken into account.

(b) Collocation method

In the collocation method the polynomial $u_N(x)$ satisfies

$$L\,u_N(x_i) = f(x_i)\,, \quad i = 1, \ldots, N-1 \tag{3.88}$$

$$B_-u_N(x_N) = g_-\,, \quad B_+u_N(x_0) = g_+\,. \tag{3.89}$$

The error equation is the equation exactly satisfied by u_N for every x. It may be written in the form

$$L\,u = f_N + p_N\,, \tag{3.90}$$

where f_N is the polynomial interpolating f at the collocation points x_i, $i = 0, \ldots, N$, and p_N is a polynomial. This polynomial p_N is of degree N at most, since $L\,u_N - f_N$ is a polynomial of degree N. From Eq.(3.88) it follows that $p_N(x_i)$ must be zero at the inner collocation points x_i, $i = 1, \ldots, N-1$, therefore it is of the form

$$p_N = (\lambda\,x + \mu)\,T'_N(x) \tag{3.91}$$

since the inner collocation points x_i are the zeros of $T'_N(x)$. The constants λ and μ in Eq.(3.91) are uniquely determined from the identity

$$L\,u_N \equiv f_N + p_N \tag{3.92}$$

since $L\,u_N - f_N$ is a known polynomial. The expressions for λ and μ may be obtained by evaluating Eq.(3.92) at two points of the continuous interval $[-1, 1]$ distinct from the collocation points. In particular, we can use the boundary points $x = \pm 1$ and we get

$$\lambda = \frac{1}{2\,N^2}\left[R_N(1) - (-1)^{N+1}R_N(-1)\right] \tag{3.93}$$

$$\mu = \frac{1}{2\,N^2}\left[R_N(1) + (-1)^{N+1}R_N(-1)\right] \tag{3.94}$$

with $R_N = L\,u_N - f_N$.

Note that the notion of error equation also applies to unsteady problems.

In a sense, the meaning of the error equation is close to the concept of the "equivalent equation" associated with finite-difference methods (Peyret and Taylor, 1983 ; Hirsch, 1988). For a finite-difference method of order p, the equivalent equation at order q, with $q > p$, is the differential equation that is satisfied by the numerical solution with an error of order q. Therefore, the approximate solution given by the scheme must reflect the properties of the equivalent equation better than those of the original differential equation. Then the study of the properties of the equivalent equation gives valuable information on the behaviour of the approximate solution. The same idea applies to the error equation since the polynomial u_N is the exact solution of an equation which is perfectly identified.

3.5 Differential equation with nonconstant coefficients

The difficulties associated with the application of the Chebyshev method to equations with nonconstant coefficients are the same as those encountered in the Fourier case (Sections 2.7 and 2.8). The arguments developed for the Fourier method also apply to the present situation.

3.5.1 Linear equation with variable coefficients

When some coefficients of Eq.(3.56) are dependent on x, the use of the Chebyshev tau method leads to difficulties analogous to those experienced with the Fourier Galerkin method. More precisely, let us assume, for example, the coefficient a in Eq.(3.56) to be nonconstant. The scalar product $(a_N\, u'_N, T_i)_w$, where a_N and u'_N are replaced by their respective Chebyshev expansions, involves a convolution sum leading to a complicated algebraic system for determining the coefficients $\hat{u}_k$ of the Chebyshev expansion of the unknown u.

Consequently, it is preferable to solve nonconstant coefficient problems by means of the collocation method described in Section 3.4.2. We obtain an algebraic system of the form (3.78) whose matrix $\mathcal{A}$ is now constructed from the matrix

$$\mathcal{Q} = -\nu\, \mathcal{D}^2 + \mathcal{D}' + b\, \mathcal{I} \tag{3.95}$$

with $\mathcal{D}' = [a(x_i)\, d^{(1)}_{i,j}]$, $i, j = 0, \ldots, N$.

3.5.2 Nonlinear equation

We consider the Burgers equation which constitutes a simple but significant model of quasi-linear unsteady problems, such as those encountered in fluid

mechanics. The Burgers equation,

$$\partial_t u + u\,\partial_x u - \nu\,\partial_{xx} u = 0\,, \tag{3.96}$$

where ν is a positive constant, has to be solved in $-1 < x < 1$ with the boundary conditions

$$u\,(-1,t) = g_-\,,\quad u\,(1,t) = g_+\,, \tag{3.97}$$

and the initial condition

$$u\,(x,0) = u_0\,(x)\ . \tag{3.98}$$

Let us denote by $u_N\,(x,t)$ the polynomial representation (with respect to x) of the solution to Eqs.(3.96)-(3.98) and by $u_N^n(x)$ the approximation of $u_N(x,t)$ at time $t_n = n\,\Delta t$, $n = 0, 1, \ldots$.

For the sake of numerical stability the time-discretization is implicit for the diffusive term $\partial_{xx} u$. As a matter of fact, an explicit evaluation of this term leads to a severe constraint on the time-step Δt which should satisfy a condition of the form $\Delta t < C/\left(\nu\,N^4\right)$, as proven in Section 4.3.2 for the heat equation. The nonlinear convective term $u\,\partial_x u$ is generally evaluated in an explicit way in order to avoid the use of an iterative procedure for the solution of the resulting algebraic system. Concerning the stability restriction associated with the explicit treatment of the convective term, it should be $\Delta t < C/\left(|u|\,N^2\right)$ in the inviscid case $\nu = 0$. More precise results are given in Chapter 4, devoted to the solution of time-dependent equations. Such a restriction on the time-step is less stringent than the one associated with the explicit evaluation of the diffusive term. Most of the spectral codes are based on the explicit treatment of the convective terms. However, in some situations, this could lead to very small time-steps and, under these conditions, it may be valuable to go further in the implicitness, even if the introduction of an iterative procedure cannot be avoided. These various possibilities are now addressed. The time-discretization is based on the simple two-level scheme already considered in the Fourier case. More accurate time-discretization schemes are discussed in Chapter 4.

(a) Explicit treatment of the convective term

Following the lines of the weighted residuals method, the residual R_N is defined by

$$R_N = \frac{u_N^{n+1} - u_N^n}{\Delta t} + u_N^n\,\partial_x u_N^n - \nu\,\partial_{xx}\,u_N^{n+1}\,. \tag{3.99}$$

This residual is set to zero at the inner collocation points x_i, $i = 1, \ldots, N-1$, and the system is closed by adding the boundary conditions, that is,

$$\frac{1}{\Delta t}\left[u_N^{n+1}(x_i) - u_N^n(x_i)\right] + u_N^n(x_i)\,\partial_x u_N^n(x_i) - \nu\,\partial_{xx} u_N^{n+1}(x_i) = 0\,,$$
$$i = 1, \ldots, N-1 \tag{3.100}$$

$$u_N^{n+1}(x_N) = g_- \,, \quad u_N^{n+1}(x_0) = g_+ \,. \tag{3.101}$$

Then the spatial derivatives $\partial_x u_N^n(x_i)$ and $\partial_{xx} u_N^{n+1}(x_i)$ are expressed, respectively, in terms of the grid values $u_N^n(x_j)$ and $u_N^{n+1}(x_j)$, $j = 0, \ldots, N$, thanks to Eq.(3.45), so that these grid values are determined by the linear algebraic system

$$\sum_{j=0}^{N} d_{i,j}^{(2)}\, u_N^{n+1}(x_j) - \sigma\, u_N^{n+1}(x_i) = \frac{1}{\nu} u_N^n(x_i)\, \partial_x u_N^n(x_i) - \sigma\, u_N^n(x_i) \,, \tag{3.102}$$

$$i = 1, \ldots, N-1$$

$$u_N^{n+1}(x_N) = g_- \,, \quad u_N^{n+1}(x_0) = g_+ \,, \tag{3.103}$$

where $\sigma = 1/(\nu \Delta t)$. Therefore, we have to solve at each time-step a system analogous to (3.78), namely,

$$\mathcal{A}\, U^{n+1} = F^n \,, \tag{3.104}$$

where $U^{n+1} = \left(u_N^{n+1}(x_0), \ldots, u_N^{n+1}(x_N)\right)^T$. The entries of the matrix $\mathcal{A}$ are constant so that it can be inverted in a precalculation stage performed before the start of the time-integration, and its inverse is stored (see Section 3.4.2). On the other hand, the forcing term F^n is changing at each time-cycle so that the efficiency of the algorithm depends partly on the way in which the nonlinear term $u_N^n(x_i)\, \partial_x u_N^n(x_i)$ is calculated. This can be done according to two different techniques.

The first technique consists of calculating the derivative $\partial_x u_N^n(x_i)$ in terms of the values $u_N^n(x_j)$, $j = 0, \ldots, N$, thanks to the differentiation matrix $\mathcal{D}$ defined by (3.51).

The second technique is based on the pseudospectral algorithm already given in the Fourier case (Section 2.8) :

1. Knowing $u_N^n(x_i)$, $i = 0, \ldots, N$, calculate by the FFT the Chebyshev coefficients $\hat{u}_k^n$, $k = 0, \ldots, N$ [from Eq.(3.37)].

2. Calculate the coefficients $\hat{u}_k^{(1)\,n}$, $k = 0, \ldots, N$, of the Chebyshev expansion of the derivative, thanks to the recurrence formula (3.28) with $p = 1$.

3. From the knowledge of the set $\hat{u}_k^{(1)\,n}$, $k = 0, \ldots, N$, calculate by the FFT the values $\partial_x u_N^n(x_i)$, $i = 0, \ldots, N$ [from Eq.(3.35)].

The efficiency of the first technique very much depends on the computer and the available matrix-vector product routines. For example, on the vector computer Cray C98, the first technique using the matrix product MXM routine, is more rapid than the second one (using the FFT) for N roughly smaller than 100. For larger values of N, the second technique becomes more rapid because of the efficiency of the FFT algorithm for large N. Finally, note that the recurrence formulas used in the second technique may be replaced by the matrix-vector product (3.24) which can be more accurate and faster.

(b) Semi-implicit treatment of the convective term

The above-mentioned constraint on the time-step, required for the stability of the explicit evaluation of the convective term, can be diminished and even avoided if this term is considered in a more implicit way. In some situations, where $u(x,t)$ can be decomposed according to $u(x,t) = \tilde{u}(x) + \overline{u}(x,t)$ with $|\tilde{u}| \geq |\overline{u}|$, we may consider the following discretization scheme

$$\frac{u_N^{n+1}(x_i) - u_N^n(x_i)}{\Delta t} + \tilde{u}_N(x_i)\,\partial_x u_N^{n+1}(x_i) - \nu\,\partial_{xx} u_N^{n+1}(x_i) = -\overline{u}_N^n(x_i)\,\partial_x u_N^n(x_i)\,, \quad i = 1,\ldots,N-1 \tag{3.105}$$

$$u_N^{n+1}(x_N) = g_-\,, \;\; u_N^{n+1}(x_0) = g_+\,, \tag{3.106}$$

which is (linearly) unconditionally stable. Moreover, because $\tilde{u}_N$ is time-independent, the resulting matrix $\mathcal{A}$ of the algebraic system is also time-independent and can be inverted once and for all in the precalculation stage. If $u(x,t)$ is positive the choice $\tilde{u} = \frac{1}{2}\max_{x,\,t}\{u(x,t)\}$ satisfies the condition $|\tilde{u}| \geq |\overline{u}|$.

(c) Implicit treatment of the convective term

In many problems, the decomposition proposed in the previous section cannot be done. The last remedy for curing the stability problem is to treat the convective term $u\,\partial_x u$ in an implicit manner. This can be done by considering either an approximation of the type $u_N^n\,\partial_x u_N^{n+1}$ or $u_N^{n+1}\,\partial_x u_N^{n+1}$. In the first case, the resulting algebraic system is linear but its coefficients are variable in time as well as in space. So, an iterative procedure has to be devised in order to avoid the inversion of the associated matrix at each time-cycle. In the second case, we are confronted with a nonlinear system which must be solved by a similar iterative procedure.

Let us now describe the algorithm associated with the first discretization mentioned above. The discrete system is

$$\frac{u_N^{n+1}(x_i) - u_N^n(x_i)}{\Delta t} + u_N^n(x_i)\,\partial_x u_N^{n+1}(x_i) - \nu\,\partial_{xx} u_N^{n+1}(x_i) = 0\,, \quad i = 1,\ldots,N-1 \tag{3.107}$$

$$u_N^{n+1}(x_N) = g_-\,, \;\; u_N^{n+1}(x_0) = g_+\,, \tag{3.108}$$

in which the spatial derivatives are expressed by means of Eq.(3.35). The system written in compact form, is

$$\mathcal{A}\,(U^n)\,U^{n+1} = F^n\,, \tag{3.109}$$

where $\mathcal{A}\,(U^n)$ is a nonlinear discrete operator. The system (3.109) is iteratively solved according to :

$$\mathcal{A}_0\,\overline{U}^{m+1} = F^n - \mathcal{A}\,(U^n)\,U^{n+1,m} \tag{3.110}$$

$$U^{n+1,m+1} = U^{n+1,m} + \alpha \overline{U}^{m+1}, \tag{3.111}$$

where m refers to the iteration. The process is initiated with $U^{n+1,0} = U_0^n$. In the above equations A is the preconditioning operator and α is the relaxation parameter.

The choice of the preconditioner $\mathcal{A}_0$ is important for the efficiency of the algorithm. It must be done according to the following requirements (Canuto *et al.*, 1988 ; Funaro, 1992 ; Quarteroni and Valli, 1994) :
(i) the matrix $\mathcal{A}_0$ must be easy to invert or, equivalently, the system of the form $\mathcal{A}_0 X = S$ must be easy to solve, and
(ii) the spectral condition number κ_{sp} of the matrix $\mathcal{A}_0^{-1}\mathcal{A}$ must be small, that is, close to 1.
For an $n \times n$ matrix $\mathcal{M}$ with eigenvalues $\lambda_i(\mathcal{M})$, $i = 1, \ldots, n$, the spectral condition number $\kappa_{sp}(\mathcal{M})$ is defined by

$$\kappa_{sp}(\mathcal{M}) = \frac{\lambda_{max}(\mathcal{M})}{\lambda_{min}(\mathcal{M})} \tag{3.112}$$

where $\lambda_{max}(\mathcal{M})$ and $\lambda_{min}(\mathcal{M})$ are, respectively, the maximal and minimal values of $|\lambda_i(\mathcal{M})|$. Note that $1 \leq \kappa_{sp}(\mathcal{M}) \leq \kappa(\mathcal{M})$ where $\kappa(\mathcal{M})$ is the usual condition number defined by Eq.(3.80). In the case where $\mathcal{M}$ is symmetric, we have $\kappa_{sp} = \kappa$.

A possible choice (Orszag, 1980 ; Canuto and Quarteroni, 1985) is to use as preconditioner $\mathcal{A}_0$ the finite-difference operator $\mathcal{A}_{df}$, analog of $\mathcal{A}$, based on the same collocation points. Therefore, the matrix $\mathcal{A}_0 = \mathcal{A}_{df}$ has also variable (in time and space) entries but is tridiagonal (for three-point formulas), so that the solution of the system (3.110) can be calculated at each time-cycle in a very efficient way, using the classical algorithm for the tridiagonal system (see, e.g., Isaacson and Keller, 1966 ; or Peyret and Taylor, 1983).

Another possible choice is the finite-element preconditioning (Canuto and Quarteroni, 1985 ; Deville and Mund, 1985, 1990 ; Quarteroni and Zampieri, 1992) based, for example, on linear elements. In this case, Eq.(3.109) is replaced by

$$\mathcal{A}_{fe}\overline{U}^{m+1} = \mathcal{M}\left[F^n - \mathcal{A}(U^n)\, U^{n+1,m}\right],$$

where $\mathcal{A}_{fe}$ is the finite-element analog to $\mathcal{A}$ and $\mathcal{M}$ is the mass matrix. For model elliptic problems (Deville and Mund, 1990 ; Quarteroni and Valli, 1994), the spectral condition number $\kappa_{sp}\left(\mathcal{A}_{fe}^{-1}\,\mathcal{M}\,\mathcal{A}\right)$ is found to be smaller than $\kappa_{sp}\left(\mathcal{A}_{df}^{-1}\mathcal{A}\right)$, thus leading to a better convergence of the iterative procedure.

Another possibility consists of using, as a preconditioner, a Chebyshev collocation operator $\mathcal{A}_0$ defined as an approximation to the operator $\mathcal{A}$ (Gauthier, 1988 ; Guillard and Désidéri, 1990 ; Fröhlich *et al.*, 1991). In the present case, $\mathcal{A}_0$ may be constructed by simply neglecting the nonconstant first-order derivative term in $\mathcal{A}$, that is, $\mathcal{A}_0 = \mathcal{A}(0)$. The resulting

matrix $\mathcal{A}_0$ is then constant and can be inverted once and for all in the precalculation stage performed previously to start the time-integration as explained in Section 3.4.2.

The choice of the relaxation parameter α is also of great importance. The use of a constant α is delicate because its optimal value may depend on U^n appearing in $\mathcal{A}$, so that it has to be changed during the time-integration. Therefore, it is preferable to use a relaxation parameter $\alpha = \alpha_m$ redefined at each iteration. This dynamic calculation of α can be done in several ways (Canuto *et al.*, 1988). Among them, the preconditioned minimal residual (PMR) method gives a satisfactory convergence at a reasonable computing time. However, in several cases (Fröhlich and Peyret, 1990 ; Sabbah *et al.*, 2001), much better convergence is obtained using the preconditioned conjugate residual (PCR) method which is a little more complicated. We refer to Section 4.3.8 for the description of the PMR and PCR methods.

3.5.3 Aliasing

The existence of aliasing in the Chebyshev approximation has been shown in Section 3.3.2. Its effect on the solution of a differential equation with nonconstant coefficients is similar to the one encountered for the Fourier method (Sections 2.7 and 2.8) due to the close relationship between Fourier and Chebyshev series. As a result, the removal of aliasing can be accomplished by again using the "3/2 rule" (Section 2.9), which is applied to the Chebyshev series expansion with obvious changes. The influence of the aliasing on the solution of the Navier-Stokes equations is discussed in Section 7.8.

3.6 Some results of convergence

The similarity between Fourier and Chebyshev series leads to similar results concerning the convergence. So, it can be shown that the rate of decay of the Chebyshev coefficient $\hat{u}_k$ defined by Eq.(3.19) is again given by Eqs.(2.7) and (2.8) (see Gottlieb and Orszag, 1977).

Concerning theoretical results on the approximation error we refer to the books by Canuto *et al.* (1988), Mercier (1989), and by Bernardi and Maday (1992) in which the following estimates are given with reference to the original papers.

Error estimates are, most of the time, obtained for functional spaces weighted with the Chebyshev weight. For example, the error of the Galerkin approximation, defined by Eqs.(3.17) and (3.19) (also called the "projection error") in the $H^p_w(-1,1)$-norm, is found to satisfy

$$\|u - u_N\|_{H^p_w(-1,1)} \leq C\, N^{-1/2+2p-m}\, \|u\|_{H^m_w(-1,1)} \tag{3.113}$$

for $1 \leq p \leq m$, if $u \in H_w^m(-1,1)$ for some $m \geq 1$. The constant C is independent of N. The space $H_w^p(-1,1)$ is the weighted Sobolev space of order p whose norm is defined by

$$\|u\|_{H_w^p(-1,1)} = \left(\sum_{k=0}^{p} \int_{-1}^{1} |u^{(k)}(x)|^2 \, w(x) \, dx \right)^{1/2} . \tag{3.114}$$

The error estimate (3.113) is nonoptimal because the power of N appearing in (3.113) is not the difference between the order of the Sobolev spaces appearing in the left- and right-hand sides of the inequality.

A norm of current use in numerical analysis is that corresponding to $p = 0$, namely, the L_w^2-norm, associated with the space of square integrable functions. For this norm, the error satisfies the inequality

$$\|u - u_N\|_{L_w^2(-1,1)} \leq C \, N^{-m} \|u\|_{H_w^m(-1,1)} . \tag{3.115}$$

This estimate is optimal. Because the error in the L_w^2-norm is of global nature and does not involve derivatives, it can be misleading in some situations where the error is not uniformly distributed and is locally large. Therefore, for practical applications, the error in the maximum norm may be of interest because it gives a finer picture of the accuracy of the approximation. Thus, for approximation (3.17), (3.19), the error estimate in the L^∞-norm is :

$$\|u - u_N\|_{L^\infty(-1,1)} \leq C \, (1 + \ln N) \; N^{-m} \sum_{k=0}^{m} \|u^{(k)}\|_{L^\infty(-1,1)} . \tag{3.116}$$

Now let us consider the collocation (i.e., "interpolation") approximation u_N defined by Eq.(3.17) with coefficients $\hat{u}_k$ given by Eq.(3.36) or, equivalently, defined by Eqs.(3.43) and (3.44). The error in the H_w^p-norm satisfies the inequality

$$\|u - u_N\|_{H_w^p(-1,1)} \leq C \, N^{2p-m} \|u\|_{H_w^m(-1,1)} \tag{3.117}$$

for $0 \leq p \leq m$. This error estimate is optimal. In the maximum norm the error is found to satisfy

$$\|u - u_N\|_{L^\infty(-1,1)} \leq C \, N^{\frac{1}{2}-m} \, \|u\|_{H_w^m(-1,1)} . \tag{3.118}$$

Therefore, the general conclusions referred to at the end of Section 2.1.2 about the accuracy of the Fourier approximation also apply to the Chebyshev approximation : exponential accuracy is obtained for infinitely differentiable functions, but this accuracy is lost for nonsmooth functions. The case where the function is discontinuous is the worst.The Gibbs oscillations make the approximation useless unless some special treatment (e.g., filtering) is done. For continuous functions which have only a finite number of

bounded derivatives, the decay of the error is only algebraic. Possible ways to solve singular problems with Chebyshev polynomial approximation are discussed in Chapter 8.

Concerning the polynomial approximation of singular functions, it is important to notice that the error decays more rapidly when the singularity is located at a boundary. The rate of convergence of the coefficients of the Chebyshev series approximating such functions has been discussed by Boyd (1986). Moreover, some theoretical results showing a doubling of the decay rate of the error have been established by Bernardi and Maday (1991) for approximations by Jacobi polynomials to functions exhibiting a singularity at an extremity of the interval.

To illustrate these properties, we now consider two examples of singular functions (compared by Botella, 1998), whose singularity is located either at an inner point of the interval or at an extremity. These functions are approximated with the collocation Chebyshev method. The first one is defined by

$$u_\alpha(x) = \begin{cases} 0 & -1 \le x < 0\,, \\ x^\alpha & 0 \le x \le 1\,, \end{cases} \tag{3.119}$$

where $\alpha > 0$. This function exhibits a singular behaviour at the center of the interval $(-1,1)$ and belongs to $H^m_w(-1,1)$ with $m < \alpha + 1/2$. Let $e_N(x)$ be the difference between the function $u_\alpha(x)$ and its interpolation polynomial $u_N(x)$, namely,

$$e_N(x) = u_\alpha(x) - u_N(x)\,.$$

Equations (3.117) and (3.118) give the following estimates

$$\begin{aligned} &\|e_N\|_{L^2_w(-1,1)} \le c_1\, N^{-1/2-\alpha}\,, \\ &\|e_N\|_{L^\infty(-1,1)} \le c_2\, N^{-\alpha}\,, \\ &\|e_N\|_{H^1_w(-1,1)} \le c_3\, N^{1/2-\alpha}\,, \end{aligned} \tag{3.120}$$

where c_1, c_2 and c_3 are positive constants independent of N. To check these estimates numerically it is necessary to evaluate the various norms with the best possible accuracy. The continuous $L^2_w(-1,1)$- and $H^1_w(-1,1)$-norms are calculated by evaluating the value of the interpolating polynomial $u_N(x)$ and its derivative $u'_N(x)$ on the $M+1$ Gauss-Lobatto points $\xi_j = \cos \pi j/M$, $j = 0, \ldots, M$, with M much larger than N. Here, $M = 1000$ for N belonging to the range $[8, 24]$. Then the integrals are evaluated by means of the Gauss-Lobatto quadrature formula (3.14) based on the points ξ_j, $j = 0, \ldots, M$. In an analogous way, the $L^\infty(-1,1)$-norm is calculated by taking the maximum of $e_N(x)$ on the above $M+1$ Gauss-Lobatto points. Table 3.3 shows the numerical estimates of the order of the error, namely, $e_N = O(N^{-q})$. The calculated value of q is in agreement with the theoretical results (3.120).

	$\alpha = 2$	$\alpha = 4$	$\alpha = 6$
$L^2_w(-1,1)$	2.5	4.5	6.5
$L^\infty(-1,1)$	2.0	4.0	6.1
$H^1_w(-1,1)$	1.5	3.5	5.6

Table 3.3. Order q of the interpolation error for the function u_α defined by Eq.(3.119).

	$\alpha = 0.9$	$\alpha = 1.9$	$\alpha = 2.9$
$L^2_w(-1,1)$	2.2	4.2	6.2
$L^\infty(-1,1)$	1.8	3.8	5.9
$H^1_w(-1,1)$	0.3	2.3	4.4

Table 3.4. Order q of the interpolation error for the function u_α defined by Eq.(3.121).

The second example concerns the function $u_\alpha(x)$ defined by

$$u_\alpha(x) = \left(1 - x^2\right)^\alpha \ , \quad -1 \leq x \leq 1 \tag{3.121}$$

where the positive constant α is not an integer. This function belongs to $H^m_w(-1,1)$ with $m < \alpha + 1/4$. The singularity is no longer located in the interior of the interval but at its extremities. Table 3.4 shows the convergence rate q of the interpolation error e_N, estimated numerically as above. From these results, the following rules can be stated

$$\begin{aligned} \|e_N\|_{L^2_w(-1,1)} &= O\left(N^{-1/2-2\,\alpha}\right) \ , \\ \|e_N\|_{L^\infty(-1,1)} &= O\left(N^{-2\,\alpha}\right) \ , \\ \|e_N\|_{H^1_w(-1,1)} &= O\left(N^{3/2-2\,\alpha}\right) \ . \end{aligned} \tag{3.122}$$

It is interesting to observe that for the L^2_w- and H^1_w-norms, the order of accuracy is exactly twice that given by the estimates (3.117) which is no longer optimal in this case where the singularity is located at an extremity. This behaviour is in agreement with the theoretical results obtained by Bernardi and Maday (1991).

The rate of decay of the coefficients of the Chebyshev series depends on the degree of smoothness of the function and on the location of the singularity. In these respects, it is instructive to compare the spectrum $|\hat{u}_k|$, $k = 0, \ldots, N$, calculated from formula (3.36), corresponding to each singular functions (3.119) and (3.121). The comparison is made on two bases : (i) the functions have the same kind of singularity characterized by $\alpha = 2.9$, and (ii) the functions belong to the same functional space H^m_w, namely $\alpha = 2.65$ for (3.119) and $\alpha = 2.9$ for (3.121). The difference in the decay rates is clearly seen in Fig.3.2.

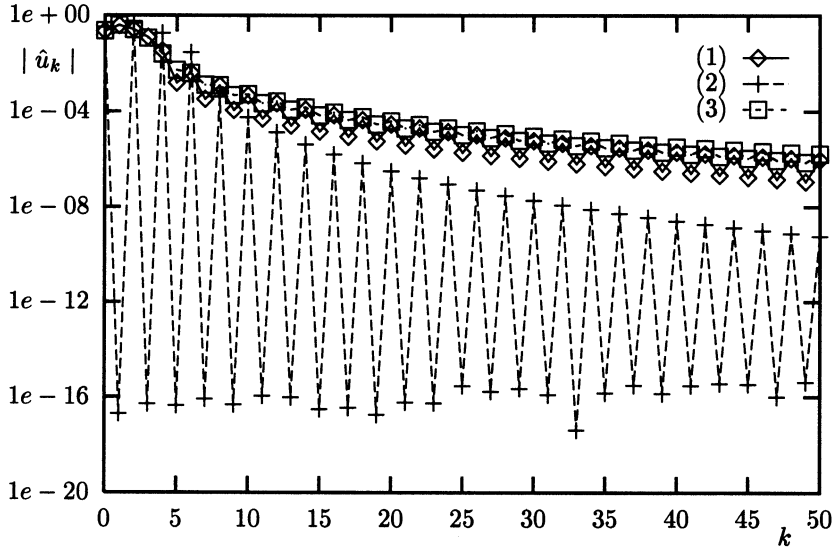

FIGURE 3.2. Chebyshev spectrum of singular functions : (1) function (3.119) with $\alpha = 2.65$, (2) function (3.119) with $\alpha = 2.9$, (3) function (3.121) with $\alpha = 2.9$.

Up to now, we have only discussed the error associated with the Chebyshev representation of a given function, either by Galerkin (projection) or collocation (interpolation) methods. When applying Chebyshev methods to calculate the solution of differential problems, the error of the approximate solution with respect to the exact one is obviously of crucial interest. Theoretical results concerning one- and two-dimensional problems are given in the books already mentioned. The nature of the error is similar to that associated with projection or interpolation errors. It depends essentially on the smoothness of the solution, the general rule being : the more the exact solution is smooth, the more the approximate solution is accurate.

However, we have to be aware that in multidimensional physically realistic problems, the solution is seldom infinitely differentiable. A very simple example is constituted by the Laplace equation in a square domain : at a corner, the intersection of two sides where homogeneous Dirichlet conditions are prescribed, the solution behaves like (Grisvard, 1985) :

$$u \sim C\, r^2\, \ln r\, \sin 2\theta\,,$$

where r and θ are polar coordinates such that r is the distance to the corner. Therefore the third-order derivatives of u are infinite at the corner and the "infinite" spectral accuracy is lost.

Another typical case is the Stokes flow in a corner with a no-slip condition. The streamfunction ψ, the solution of the biharmonic equation, behaves like (Moffatt, 1964) :

$$\psi \sim C\, r^{3.74} \left[\cos\left(1.13\, \ln r\right)\, g(\theta) + \sin\left(1.13\, \ln r\right)\, h(\theta)\right] \tag{3.123}$$

so that the second-order derivatives of the velocity become infinite at the corner. We refer to Chapter 8 for a more general discussion on the treatment of singularities.

For now, numerical examples (Botella, 1998) are displayed in order to illustrate the accuracy of spectral methods compared to finite-difference methods in the case of regular and (weakly) singular solution. More precisely, we consider the solution of the one-dimensional Helmholtz equation

$$-\nu\, u'' + u = f\,, \quad -1 < x < 1\,, \quad \nu = 10^{-2} \tag{3.124}$$

with

$$u(-1) = g_-\,, \quad u(1) = g_+\,, \tag{3.125}$$

using various methods :

(1) Second-order finite-difference method with uniform mesh ($\Delta x = 2/N$).
(2) Second-order finite-difference method with Gauss-Lobatto mesh (3.75).
(3) Sixth-order Hermitian method with uniform mesh ($\Delta x = 2/N$).
(4) Sixth-order Hermitian method with Gauss-Lobatto mesh (3.75).
(5) Chebyshev collocation method [Gauss-Lobatto mesh (3.75].
(6) Chebyshev tau method (polynomial degree = N).

The Hermitian method is based on the formula using three points for the second-order derivative and five points for the function (Collatz, 1966 ; Botella, 1988 ; Peyret, 2000). The collocation method has been described in Section 3.4.2 and the resulting algebraic system is solved by the diagonalization technique (Section 3.7.1). The tau method applied to the Helmholtz equation (Section 3.4.1) leads to the solution of quasi-tridiagonal systems obtained through the algorithm described in Appendix B.

The error is measured with the discrete L^2-norm defined by

$$\overline{E} = \left[\frac{1}{N-1} \sum_{j=1}^{N-1} |u_i - u(x_i)|^2\right]^{1/2}\,, \tag{3.126}$$

where u_i refers to the approximate solution and $u(x_i)$ to the exact one.

Figure 3.3.a shows the error $\overline{E}$ found for the smooth solution

$$u(x) = 1 - \frac{\sinh\left[(x+1)\,/\sqrt{\nu}\right]}{\sinh\left(2/\sqrt{\nu}\right)} \tag{3.127}$$

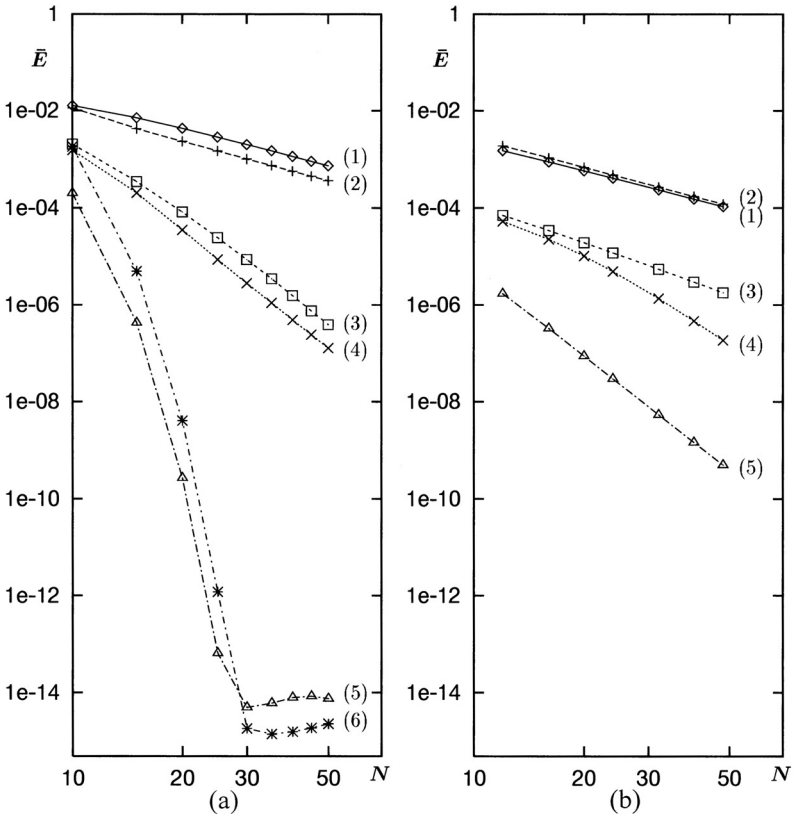

FIGURE 3.3. Error E versus the number N of degrees of freedom : (a) smooth solution (3.127), (b) singular solution (3.121) (see the text for the meaning of the labels).

obtained for $f = 1$, $g_- = 1$, and $g_+ = 0$. The exponential accuracy of the spectral methods is clearly seen in this figure. For $N < 30$ the error of the collocation method is smaller than the error of the tau method. For larger values of N, in the range of round-off errors, the accuracy of the tau method is better. This is a consequence of the ill-conditioning of the collocation matrix, although its effect is largely reduced by the use of the matrix-diagonalization method.

The second example concerns the calculation of a solution exhibiting a singularity at the boundaries $x = \pm 1$. Therefore, the computed solution is $u(x) = u_\alpha(x)$ given by Eq.(3.121), which defines f in (3.124) and $g_- = g_+ = 0$. Figure 3.3.b shows the error $\overline{E}$ in the case $\alpha = 2.9$. Results concerning the order q of the error $\overline{E} = O(N^{-q})$ for various values of α are given in Table 3.5. Some interesting conclusions can be drawn from these numerical results.

	Sixth-order Hermitian		Chebyshev
α	Uniform mesh	Gauss-Lobatto mesh	collocation
2.1	2.1	4.2	4.7
2.9	2.9	5.6	5.9
3.1	3.1	5.9	6.3
4.9	5.2	6.0	10.5

Table 3.5. Order q of the error $\overline{E}$ (discrete L^2-norm) associated with the solution (3.121).

First, it is observed that, in agreement with the general theory, the error of the Chebyshev collocation method (curve (5)) is no longer exponential but algebraic. From an evaluation of the error of the numerical solution in various continuous norms as done previously, it is found that the behaviour of this error again follows the rules stated in (3.122), namely, the same behaviour as the interpolation error.

Second, it is seen that the error of the Hermitian method is, at least in the considered range of N, $O(N^{-2.9})$ for the uniform mesh (curve (3)) and $O(N^{-5.6})$ for the variable Gauss-Lobatto mesh (curve (4)). This shows how the singular nature of the solution may have an effect on the rate of convergence of a high-order finite-difference method when the resolution near the singular point is not sufficient. Moreover, although the respective rates of convergence of the Hermitian and Chebyshev methods are close (curves (4) and (5)), the magnitude of the error of the Chebyshev method is much smaller. Therefore, in problems with weak singularities like those considered here, the Chebyshev collocation method is more efficient than the sixth-order Hermitian method, as much as the implementation of high-order Hermitian methods in the variable mesh may be complicated.

3.7 Multidimensional elliptic equation

The Chebyshev methods (tau or collocation) which have been described in the previous sections apply straightforwardly to multidimensional equations, provided the computation domain is a square or a cube. However, the resulting algebraic systems have to be solved by a more efficient technique than the direct inversion method proposed for the one-dimensional case.

The solution method usually employed, which is found to be very efficient in unsteady problems where the same equation has to be solved a large number of times, is the matrix-diagonalization procedure. This method was introduced by Lynch *et al.* (1964) for finite-difference approximation. It was applied to the Chebyshev tau method by Haidvogel and Zang (1979) in the two-dimensional case and by Haldenwang *et al.* (1984) to the three-

dimensional case. The application to the collocation method was considered by Ehrenstein and Peyret (1989).

In the present section, we consider the solution of the Helmholtz equation, but the method applies to more general problems provided they are separable and the associated one-dimensional matrices are diagonalizable. This last point requires that the eigenvector matrices are well-conditioned. This requirement may not be met for the steady advection-diffusion equation and, in this case, the Schur-decomposition method constitutes an efficient way to solve the associated algebraic system.

First, in order to give an idea about its basic principles, the matrix-diagonalization method is presented for the solution of the one-dimensional Helmholtz equation. However, more general operators, for which some eigenvalues may be complex, will also be discussed. The case of the steady-state advection-diffusion equation will be considered and the Schur-decomposition method will be discussed and compared to the matrix-diagonalization method. Then the solution of the two-dimensional Helmholtz equation will be considered in detail ; in particular, the case where the coefficients, that occur in the general boundary conditions of Robin type, vary on the same side of the boundary will be addressed. Finally, the algorithm for the three-dimensional equation will be described.

Only the collocation approximation will be considered here, since the application to the tau method is similar (see the above references).

3.7.1 One-dimensional equation

Let us consider the one-dimensional Helmholtz equation with general Robin boundary conditions

$$u'' - \sigma\, u = f\,, \quad -1 < x < 1 \tag{3.128}$$

$$\alpha_-\, u(-1) + \beta_-\, u'(-1) = g_- \tag{3.129}$$

$$\alpha_+\, u(1) + \beta_+\, u'(1) = g_+\,, \tag{3.130}$$

where σ is a positive constant. It is assumed that $\alpha_-\alpha_+ \geq 0$ without loss of generality.

The application of the collocation method (Section 3.4.2) leads to an algebraic system for the unknowns $u_N(x_i)$, $i = 0, \ldots, N$. For the diagonalization procedure, it is convenient to eliminate the boundary values, thanks to the boundary conditions (3.129) and (3.130) so that the unknown vector is

$$V = (u_N(x_1), \ldots, u_N(x_{N-1}))^T\,. \tag{3.131}$$

The boundary values $u_N(x_0)$ and $u_N(x_N)$ are expressed from Eqs.(3.129) and (3.130) by

$$u_N(x_0) = \frac{1}{e}\sum_{j=1}^{N-1} b_{0,j} u_N(x_j) + \frac{1}{e}\left(c_{0,-}g_- + c_{0,+}g_+\right)\,, \tag{3.132}$$

$$u_N(x_N) = \frac{1}{e}\sum_{j=1}^{N-1} b_{N,j} u_N(x_j) + \frac{1}{e}\left(c_{N,-}g_- + c_{N,+}g_+\right) , \tag{3.133}$$

with

$$\begin{aligned}
&e = c_{0,+}c_{N,-} - c_{0,-}c_{N,+} ,\\
&c_{0,-} = -\beta_+ d^{(1)}_{0,N} , \;\; c_{0,+} = \alpha_- + \beta_- d^{(1)}_{N,N} ,\\
&c_{N,+} = -\beta_- d^{(1)}_{N,0} , \;\; c_{N,-} = \alpha_+ + \beta_+ d^{(1)}_{0,0} ,\\
&b_{0,j} = -c_{0,+}\beta_+ d^{(1)}_{0,j} - c_{0,-}\beta_- d^{(1)}_{N,j} , \qquad j = 1,\ldots,N-1 ,\\
&b_{N,j} = -c_{N,-}\beta_- d^{(1)}_{N,j} - c_{N,+}\beta_+ d^{(1)}_{0,j} , \qquad j = 1,\ldots,N-1 .
\end{aligned} \tag{3.134}$$

It is recalled that $d^{(1)}_{i,j}$ and $d^{(2)}_{i,j}$ are the elements of the differentiation matrices [see Eq.(3.46) and Eq.(3.47)].

The discrete system approximating (3.128)-(3.130) can be written as

$$\left(\mathcal{D}_x - \sigma\,\mathcal{I}\right) V = H , \tag{3.135}$$

where $\mathcal{I}$ is the $(N-1)\times(N-1)$ identity matrix and the matrix $\mathcal{D}_x = [d_{i,j}]$, $i,j = 1,\ldots,N-1$, is defined by

$$d_{i,j} = d^{(2)}_{i,j} + \frac{1}{e}\left(b_{0,j}\, d^{(2)}_{i,0} + b_{N,j}\, d^{(2)}_{i,N}\right)$$

and $H = [h_i]$, $i = 1,\ldots,N-1$, is the vector such that

$$h_i = f(x_i) - \frac{1}{e}\left(c_{0,-}g_- + c_{0,+}g_+\right) d^{(2)}_{i,0} - \frac{1}{e}\left(c_{N,-}g_- + c_{N,+}g_+\right) d^{(2)}_{i,N} .$$

The algorithm is based on the diagonalization of the matrix $\mathcal{D}_x$. It has been proven by Gottlieb and Lutsman (1983) that the eigenvalues λ_i, $i = 1,\ldots,N-1$, of $\mathcal{D}_x$ are real, negative and distinct if the following conditions are satisfied

$$\alpha_- > 0 , \;\; \beta_- < 0 , \;\; \alpha_+ > 0 , \;\; \beta_+ > 0 . \tag{3.136}$$

These conditions can be relaxed to include the cases $\beta_- = \beta_+ = 0$ (Dirichlet-Dirichlet), $\beta_- = \alpha_+ = 0$ (Dirichlet-Neumann), and $\alpha_- = \alpha_+ = 0$ (Neumann-Neumann). In this latter case, one eigenvalue is zero.

For the situations specified above the matrix $\mathcal{D}_x$ has $N-1$ linearly independent eigenvectors such that it can be diagonalized and

$$\mathcal{D}_x = \mathcal{P}\Lambda\mathcal{P}^{-1} , \tag{3.137}$$

where Λ is the diagonal matrix whose entries are the eigenvalues λ_i, $i = 1,\ldots,N-1$, the transformation matrix $\mathcal{P}$ is the matrix whose columns

are the eigenvectors and $\mathcal{P}^{-1}$ is its inverse. Therefore, Eq.(3.135) may be written as

$$\left(\mathcal{P}\,\Lambda\,\mathcal{P}^{-1} - \sigma\mathcal{I}\right) V = H$$

and, after left multiplication by $\mathcal{P}^{-1}$, we get

$$\left(\Lambda - \sigma\,\mathcal{I}\right)\mathcal{P}^{-1}\,V = \mathcal{P}^{-1}\,H\,.$$

Then, with $\tilde{V} = \mathcal{P}^{-1}V$ and $\tilde{H} = \mathcal{P}^{-1}H$, this equation becomes

$$\left(\Lambda - \sigma\,\mathcal{I}\right)\tilde{V} = \tilde{H}$$

and

$$\tilde{V} = \left(\Lambda - \sigma\mathcal{I}\right)^{-1}\tilde{H}\,. \tag{3.138}$$

The matrix $\Lambda - \sigma\mathcal{I}$ is diagonal, such that it is easily inverted. Its inverse is simply the diagonal matrix with entries equal to $1/\left(\lambda_i - \sigma\right)$. Because $\lambda_i \leq 0$, these quantities are not zero provided $\sigma > 0$.

The general algorithm is as follows :

1. Calculate $\tilde{H} = \mathcal{P}^{-1}H$.
2. Calculate $\tilde{V} = \left(\Lambda - \sigma\mathcal{I}\right)^{-1}\tilde{H}$.
3. Calculate $V = \mathcal{P}\tilde{V}$.
4. Calculate the boundary values $u_N(x_0)$ and $u_N(x_N)$.

In the special case of the Neumann problem for the Poisson equation ($\sigma = 0$), the matrix $\mathcal{D}_x$ has one eigenvalue equal to zero. In the calculation of the vector $\tilde{V}$, the peculiar component of $\tilde{V}$ corresponding to this null eigenvalue is simply taken arbitrarily equal to zero. This arbitrariness is consistent with the fact that the solution of the Neumann problem for the Poisson equation is defined up to a constant (obviously when the compatibility condition is satisfied).

The computational effort with this algorithm is obviously much heavier than those required by the direct inversion of $\mathcal{A} = \mathcal{D}_x - \sigma\mathcal{I}$. Even in the case of unsteady problems, for which the calculation of the λ_i, $\mathcal{P}$, and $\mathcal{P}^{-1}$ may be done once and for all in the preprocessing stage performed before the start of the time-integration, the algorithm remains more costly. However, it becomes really efficient for multidimensional problems as will be seen below.

The use of the matrix-diagonalization method necessitates the calculation of the eigenvalues of the matrix $\mathcal{D}_x$ of the associated eigenvectors and the inversion of the eigenvector matrix $\mathcal{P}$. It is known (e.g., Canuto *et al.* 1988) that the largest eigenvalues of $\mathcal{D}_x$ vary like N^4 as $N \to \infty$, so that these eigenvalues are not necessarily good approximations to the eigenvalues of the continuous operator (Weidman and Trefethen, 1988). On the other hand, the effect of round-off errors on the calculation of the eigenvalues of the second-order differentiation operator is negligible (contrary

to the case of the first-order derivative). This is due to the fact that the sensitivity of the eigenvalues is measured by the condition number of the eigenvector matrix $\mathcal{P}$ (and not $\mathcal{D}_x$), as stated by the Bauer-Fike theorem (Wilkinson, 1965). In the present case, the matrix $\mathcal{P}$ is well-conditioned. For example, the condition number $\kappa(\mathcal{D}_x)$ in the Dirichlet case behaves like $0.80N^{1/4}$ (Ehrenstein, 1986). This property also ensures that no numerical difficulties are encountered when inverting the matrix $\mathcal{P}$.

Remarks

1. *Case of variable σ*
In the Helmholtz equation (3.128) the coefficient σ is assumed to be a constant. However the solution technique described above also applies if σ is a function time t, when the Helmholtz equation to be solved is the result of a time-discretization process. If σ depends on x, we diagonlize the matrix associated with u''/σ or $u'' - \mu\sigma$.

2. *Case of complex eigenvalues*
For elliptic equations different from the Helmholtz equation (3.128), such as for equations resulting from the application of a coordinate transformation to (3.128) or else for the advection-diffusion equation, the corresponding matrix $\mathcal{D}_x$ may have (conjugate) complex eigenvalues. In such a case, it is possible to avoid complex number computations, owing to the quasi-diagonalization procedure considered by Pasquetti and Bwemba (1994) for the solution of the equation $u'' + (2m+1)u'/x = f$ (m = integer) coming from the Poisson equation in a cylindrical coordinate system.

For the presentation of the technique, one assumes that $\mathcal{D}_x$ has two conjugate complex eigenvalues λ_1 and $\lambda_2 = \overline{\lambda}_1$, the associated eigenvectors being W_1 and $\overline{W}_1$. Now let us introduce the following partition of the matrices Λ and $\mathcal{P}$:

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda' \end{bmatrix}, \quad \mathcal{P} = [\mathcal{P}_1 \quad \mathcal{P}'] ,$$

where Λ_1 is the 2×2 matrix

$$\Lambda_1 = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \bar{\lambda}_1 \end{bmatrix}$$

and $\mathcal{P}_1$ is the $(N-1) \times 2$ matrix constructed with the eigenvectors W_1 and $\overline{W}_1$:

$$\mathcal{P}_1 = \begin{bmatrix} W_1 & \overline{W}_1 \end{bmatrix}$$

and Λ', $\mathcal{P}'$ being the complementary parts. It is easy to show that the real matrices $\mathcal{K}$ and $\mathcal{T}$ respectively defined by

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_1 & 0 \\ 0 & \Lambda' \end{bmatrix}, \quad \mathcal{T} = [\mathcal{T}_1 \quad \mathcal{P}'] ,$$

with

$$\mathcal{K}_1 = \begin{bmatrix} \mathcal{R}e(\lambda_1) & \mathcal{I}m(\lambda_1) \\ -\mathcal{I}m(\lambda_1) & \mathcal{R}e(\lambda_1) \end{bmatrix}, \quad \mathcal{T}_1 = [\mathcal{R}e(W_1) \quad \mathcal{I}m(W_1)] ,$$

are such that

$$\mathcal{D}_x = \mathcal{T}\,\mathcal{K}\,\mathcal{T}^{-1} . \tag{3.139}$$

Therefore, this equation replaces Eq.(3.137) and the algorithm described above can be applied in the same way except that the determination of $\tilde{V}$, as the solution of

$$(\mathcal{K} - \sigma\mathcal{I})\,\tilde{V} = \tilde{H} , \tag{3.140}$$

no longer leads to uncoupled equations : two of them (corresponding to $\mathcal{K}_1$) are coupled. This technique obviously applies to more than one couple of complex eigenvalues : each couple has to be replaced by the real 2×2 matrix like $\mathcal{K}_1$ to constitute the matrix $\mathcal{K}$, as well as the associated eigenvectors are replaced by their real and imaginary parts as in $\mathcal{T}_1$ to constitute the matrix $\mathcal{T}$. The resulting system (3.140) yields a system of uncoupled equations (for the real eigenvalues) and at most coupled two-by-two (for each couple of conjugate complex eigenvalues).

3. *Steady-state advection-diffusion equation*
In this remark, we want to point out the difficulty associated with the application of the matrix-diagonalization method to the solution of the algebraic system resulting from the discretization of equation (3.56) with $a \neq 0$, and its multidimensional analog. Then a better adapted solution method, namely, the Schur-decomposition method, will be described and applied to two typical examples.

When $a \neq 0$ equation (3.56) is of advection-diffusion type. Such an equation arises in the course of the solution of the time-dependent advection-diffusion equation discretized with an implicit scheme [see Eq.(4.49)]. It may also appear in the solution of nonlinear equations like the Burgers equation or the Navier-Stokes equations when the nonlinear first-order derivative term is approximated in a semi-implicit way as described in Section 3.5.2 [see also Eq.(4.53)]. Such a semi-implicit technique has been used by Forestier *et al.* (2000a,b) for the calculation of wake flows.

We consider the steady-state advection-diffusion equation

$$u'' - a\,u' - \sigma\,u = f , \quad -1 < x < 1 \tag{3.141}$$

where the constant coefficients a and σ are positive. Note that a could be negative without any influence on the conclusions. The Dirichlet boundary conditions

$$u\,(-1) = g_- , \quad u\,(1) = g_+ , \tag{3.142}$$

are associated with Eq.(3.141), but more general conditions could be considered as well.

The problem (3.141)-(3.142) is approximated with the Chebyshev collocation method as described above in this section. So, the algebraic system determining the unknowns $u_N(x_i)$, $i = 1, \cdots, N-1$, has the form

$$(\mathcal{A} - \sigma\,\mathcal{I})\,V = H\,, \tag{3.143}$$

where V is the vector of inner unknowns [Eq.(3.131)] and $\mathcal{A}$ is a $(N-1) \times (N-1)$ matrix constructed from the Chebyshev differentiation matrix.

The eigenvalues of the matrix are generally complex (see Section 4.2.3 for some properties of these eigenvalues ; see also Nana Kouamen, 1992 ; Reddy and Trefethen, 1994). Therefore, the modification of the matrix-diagonalization method given above in Remark 2 has to be employed. Unfortunately, the transformation matrix $\mathcal{T}$ [see Eq.(3.139)] may be ill-conditioned for some values of a and N (see Fig.3.3). In such a situation, the inversion of $\mathcal{T}$ is inaccurate or even impossible.

This behaviour may be explained by examining the eigenvalue problem associated to the advection-diffusion operator, namely,

$$\varphi'' - a\,\varphi' = \lambda\,\varphi\,, \quad -1 < x < 1 \tag{3.144}$$

$$\varphi\,(-1) = 0\,, \quad \varphi\,(1) = 0\,. \tag{3.145}$$

The eigenvalues λ_n and the associated eigenvectors φ_n are, respectively,

$$\lambda_n = -\frac{a^2}{4} - \frac{\pi^2\,n^2}{4}\,, \quad n = 1, 2, 3, \cdots\,, \tag{3.146}$$

and

$$\varphi_n = \begin{cases} e^{a\,x/2} \cos\dfrac{\pi\,n}{2} x & \text{if } n \text{ is odd}\,, \\ \\ e^{a\,x/2} \sin\dfrac{\pi\,n}{2} x & \text{if } n \text{ is even.} \end{cases} \tag{3.147}$$

As stated by Reddy and Trefethen (1994), the condition number κ_C of the basis of eigenfunctions, normalized such that $\|\varphi_n\| = 1$, is given by

$$\kappa_C = e^a\,. \tag{3.148}$$

Therefore, κ_C is exponentially increasing with a. We may expect the condition number κ_N of the transformation matrix $\mathcal{T}$ to have similar behaviour, at least if N is sufficiently large to ensure an accurate representation of the eigenfunctions. Figure 3.4 compares the variations of κ_C and κ_N (more precisely, their inverses) in terms of a. For a fixed value of N, κ_N follows κ_C when a is small ; then, for larger values of a, κ_N deviates from κ_C and reaches much smaller values. Thus, for any value of N, there exists a range of a where the ill-conditioning of $\mathcal{T}$ makes inaccurate, and even impossible, the use of the matrix-diagonalization method.

A simple remedy may be to set $v = \exp(-ax/2)\;u$ in Eq.(3.141) so as to get a Helmholtz equation [as done by Shizgal (2002) for another purpose],

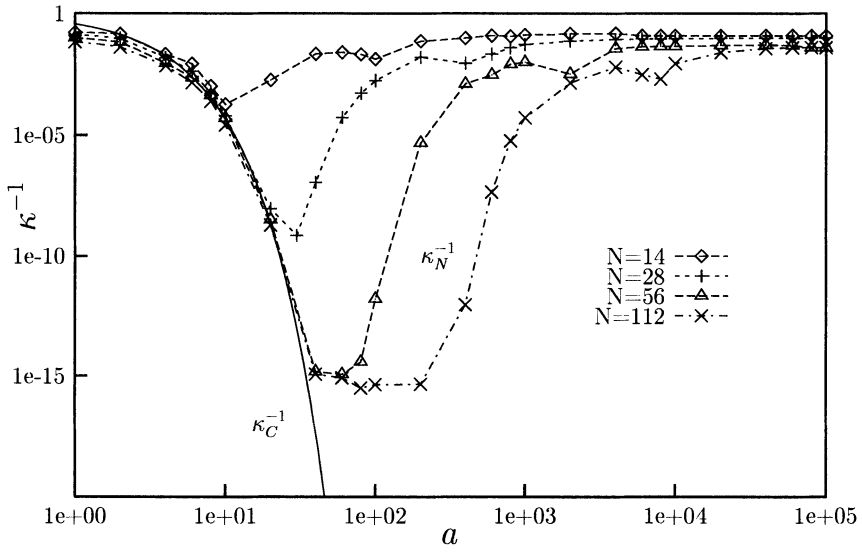

FIGURE 3.4. Dependence on a of κ_C^{-1} and κ_N^{-1} for various values of N.

accurately solved by the matrix-diagonalization technique. However, the numerical error associated with the solution of the transformed problem, which becomes stiff when a is large, restricts the application of this change of variables to relatively small values of a.

An efficient solution method of more general application is constituted by the Schur-decomposition method. The method, developed by Barthels and Stewart (1971), is based on the Schur reduction of the matrix $\mathcal{A}$ to triangular real form by orthogonal transformation. More precisely, the matrix $\mathcal{A}$ is transformed according to

$$\mathcal{A}' = \mathcal{P}^T \mathcal{A} \mathcal{P}\,, \tag{3.149}$$

where $\mathcal{P}$ is an orthogonal matrix. The real matrix $\mathcal{A}'$ is block-upper-triangular

$$\mathcal{A}' = \left[\mathcal{A}'_{i,j}\right]\,, \quad i,j = 1,\cdots,p\,,$$

with the matrix $\mathcal{A}'_{i,j} = 0$ for $j < i$ and $p = N - 2q$, where $2q$ is the number of complex eigenvalues. Each matrix $\mathcal{A}'_{i,j}$ is of order at most 2 because of the presence of these complex eigenvalues. To be more precise, let us assume that λ_1, $\lambda_2 = \overline{\lambda}_1$ are two conjugate complex eigenvalues. Then, the 2×2 matrix $\mathcal{A}'_{1,1}$ has the form

$$\mathcal{A}'_{1,1} = \begin{pmatrix} \mathcal{R}e\,(\lambda_1) & a'_{1,2} \\ a'_{2,1} & \mathcal{R}e\,(\lambda_1) \end{pmatrix}\,,$$

where $a'_{1,2}\,a'_{2,1} < 0$ and such that $\sqrt{|a'_{1,2}\,a'_{2,1}|} = |\mathcal{I}m\,(\lambda_1)\,|$.

The matrices $\mathcal{A}'$ and $\mathcal{P}$ are calculated by means of library routines (e.g., SGEES from LAPACK ; see also Barthels and Stewart, 1971). Then the solution algorithm follows that given above for the matrix-diagonalization method.

First, from Eq.(3.149), we have $\mathcal{A} = \mathcal{P}\,\mathcal{A}'\,\mathcal{P}^T$ that we bring into Eq.(3.143) which becomes

$$\left(\mathcal{P}\,\mathcal{A}'\,\mathcal{P}^T - \sigma\,\mathcal{I}\right)\,V = H\,.$$

Then, after left multiplication by $\mathcal{P}^T$, we obtain

$$\left(\mathcal{A}' - \sigma\,\mathcal{I}\right)\tilde{V} = \tilde{H}\,, \tag{3.150}$$

where $\tilde{V} = \mathcal{P}^T V = [\tilde{u}_N\,(x_i)]$, $i = 1, \cdots, N-1$, and $\tilde{H} = \mathcal{P}^T H$. The matrix

$$\mathcal{A}' - \sigma\,\mathcal{I} = \left[a'_{i,j} - \sigma\,\delta_{i,j}\right]\,, \quad i,j = 1, \cdots, N-1\,,$$

is quasi-upper-triangular. The system (3.150) is solved by back-substitution. The unknowns $\tilde{u}_N\,(x_i)$ are calculated explicitly for real eigenvalues but their determination necessitates the solution of a 2×2 system for each couple of conjugate complex eigenvalues.

To resume, the algorithm is as follows :

1. Calculate $\tilde{H} = \mathcal{P}^T H$.
2. Solve $\left(\mathcal{A}' - \sigma\,\mathcal{I}\right)\tilde{V} = \tilde{H}$.
3. Calculate $V = \mathcal{P}\tilde{V}$.

Like the matrix-diagonalization method, the Schur-decomposition method is of interest for the solution of multidimensional problems arising at each time-cycle of a time-dependent process. The calculation of $\mathcal{A}'$ and $\mathcal{P}$ is done once and for all before the start of the time-integration. Then the above algorithm is applied at each time-cycle. Note that point 2 is more expensive than point 2 of the matrix-diagonalization algorithm.

The extension to multidimensional problems is analogous to that described in the following sections for the matrix-diagonalization method. Note that both methods can be applied in combination, for example, to solve the equation

$$\left(\partial_{xx} - a\,\partial_x\right)u + \partial_{yy}u - \sigma\,u = f\,, \tag{3.151}$$

Schur-decomposition is associated with the operator $\partial_{xx} - a\,\partial_x$ and matrix-diagonalization with ∂_{yy}.

Now we present two examples of application (Forestier, 2000). The first solution is very smooth and is well-represented by a polynomial of relatively low degree whatever a. The second solution is stiff and exhibits a boundary layer for large values of a. These examples allow us to compare the Schur-decomposition and matrix-diagonalization methods. The discussion is based on the level of satisfaction of the two usual requirements for a numerical solution :

1. The algebraic system accurately represents the continuous problem.
2. The solution algorithm accurately solves this system.

Both solutions are calculated with $N = 28$ and by making $\sigma = a$ in Eq.(3.141). The numerical error E is measured by the discrete L^∞-error defined by

$$E = \max_i |u_i - u(x_i)| , \tag{3.152}$$

where u_i refers to the approximate solution and $u(x_i)$ to the exact one.

The first example concerns the solution

$$u = \sin 2\pi (x + 1) \tag{3.153}$$

that defines f in Eq.(3.141) and $g_\pm = 0$ in Eq.(3.142). The function (3.153) is independent of a and is perfectly represented with $N = 28$. Figure 3.5 displays the errors in terms of a given by the Schur-decomposition method [curve (1)] and by the matrix-diagonalization [curve (2)] method. The inverse κ_N^{-1} of the condition number of the transformation matrix $\mathcal{T}$ involved in the diagonalization procedure is also shown in Fig.3.5. The error of the Schur-decomposition method is 10^{-14} (i.e. zero at the machine precision) whatever a. This proves, by the way, that requirement (1) is well satisfied with $N = 28$ for any value of a. Also, requirement (2) is satisfied by the Schur-decomposition method. On the other hand, requirement (2) is not satisfied for every value of a by the matrix-diagonalization method. As announced, it may be observed that the accuracy of the method is directly connected to the conditioning of the transformation matrix $\mathcal{T}$.

The second example of solution is

$$u = \frac{1}{e^{2(r_2 - r_1)} - 1} \left[e^{2r_2 + r_1(x-1)} - e^{r_2(x+1)} \right] \tag{3.154}$$

with

$$r_1 = \frac{a - \sqrt{\Delta}}{2} , \quad r_2 = \frac{a + \sqrt{\Delta}}{2} , \quad \Delta = a(a + 4) .$$

This function is the solution to problem (3.141)-(3.142) with $\sigma = a$, $f = 0$, $g_- = 1$, and $g_+ = 0$. For large values of a, it exhibits a boundary layer near $x = 1$ whose thickness is $O\left(a^{-1}\right)$. In this case, requirement (1) is satisfied only if N is sufficiently large. Thus, for large a, we may expect the solution to be inaccurate for $N = 28$ whatever the algorithm used to solve the algebraic system. This is seen in Fig.3.6. As previously, the matrix-diagonalization method gives an error much larger than the Schur-decomposition method when κ_N is large [requirement (2) is not satisfied].

Lastly, it must be noticed that increasing N improves the error of the Schur-decomposition method. On the other hand, the error associated with the matrix-diagonalization method increases because the condition number κ_N deteriorates since it tends toward the continuous one κ_C when N increases (Fig.3.4).

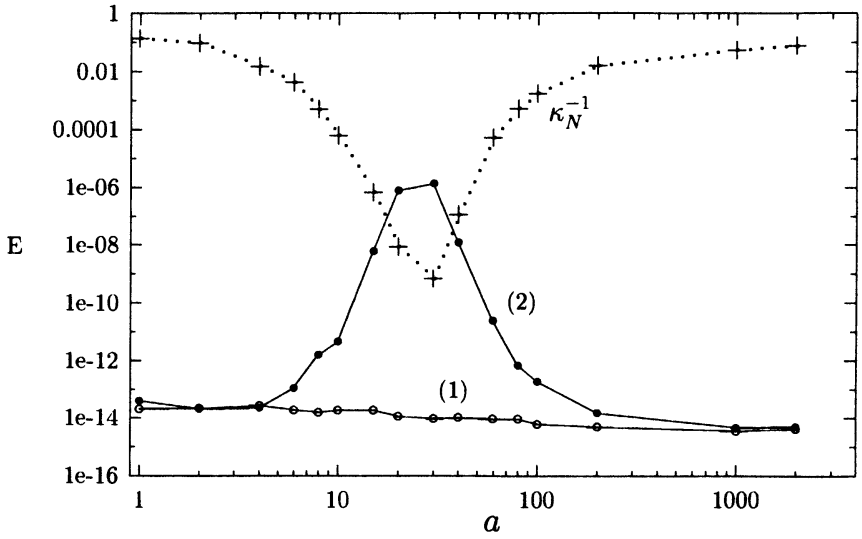

FIGURE 3.5. Error E versus a for the solution (3.153) calculated with $N = 28$: (1) Schur-decomposition method, (2) Matrix-diagonalisation method. The figure also shows the variation of κ_N^{-1}.

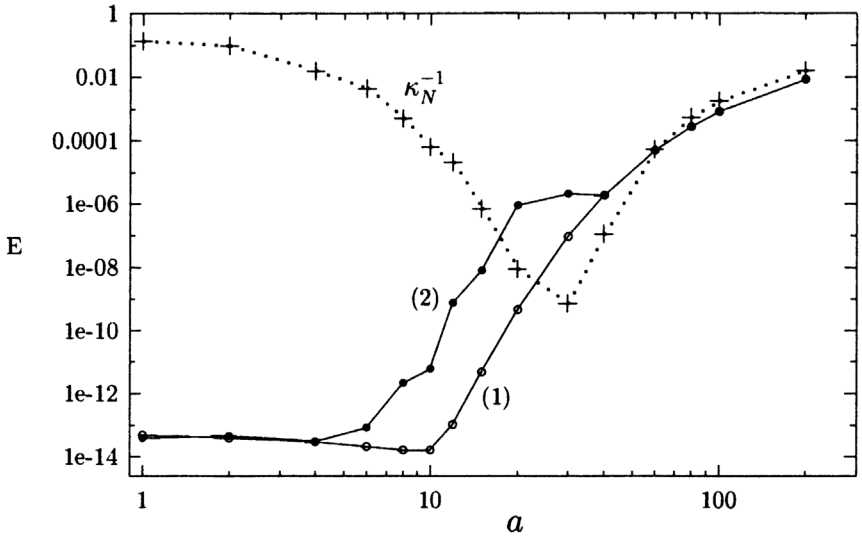

FIGURE 3.6. Error E versus a for the solution (3.154) calculated with $N = 28$: (1) Schur-decomposition method, (2) Matrix-diagonalization method. The figure also shows the variation of κ_N^{-1}.

3.7.2 Two-dimensional equation

In this section we describe the matrix-diagonalization procedure for solving the algebraic system resulting from the collocation approximation to the two-dimensional Helmholtz equation with Robin boundary conditions. The problem to be solved is

$$\partial_{xx}u + \partial_{yy}u - \sigma\, u = f \quad \text{in } \Omega \tag{3.155}$$

$$\alpha\, u + \beta\, \partial_n u = g \quad \text{on } \Gamma = \partial\Omega\,, \tag{3.156}$$

where Ω is the square $(-1,1)^2$, Γ is its boundary, and ∂_n refers to the normal derivative to Γ. For simplicity, the coefficient σ is assumed to be a constant ; the case of variable σ has been mentioned in Remark 1 of Section 3.7.1. It is also assumed that α and β are constant on each side of the square Ω but they may vary from one side to the other. The case where α and β vary along the same side will be considered in Remark 1.

The solution $u(x,y)$ is approximated with the polynomial $u_N(x,y)$ of degree at most equal to N_x in the x-direction and N_y in the y-direction. The Chebyshev collocation approximation to the problem (3.155)-(3.156) makes use of the Gauss-Lobatto mesh $\overline{\Omega}_N$ defined by

$$x_i = \cos\frac{\pi i}{N_x}\,,\quad i = 0,\ldots,N_x, \qquad y_j = \cos\frac{\pi j}{N_y}\,,\quad j = 0,\ldots,N_y\,.$$

We denote by Ω_N the open discretized domain $\Omega_N = \{x_i, y_j\}$, $i = 1,\ldots,$ $\overline{N}_x$, $j = 1,\ldots,\overline{N}_y$ where $\overline{N}_x = N_x - 1$ and $\overline{N}_y = N_y - 1$. Lastly, Γ_N^I is the discretized boundary without the four corners.

The various derivatives in each direction are approximated with expressions analogous to Eq.(3.45). Equation (3.155) is forced to be satisfied at the inner collocation points $(x_i, y_j) \in \Omega_N$ and the boundary condition (3.156) is applied at the inner boundary points $(x_i, y_j) \in \Gamma_N^I$.

As in the one-dimensional case considered in the previous section, the boundary values $u_N(x_i, y_j)$ for $(x_i, y_j) \in \Gamma_N^I$ are eliminated from the global algebraic system thanks to the boundary conditions (3.156), giving equations similar to (3.132) and (3.133).

For an efficient application of the matrix-diagonalization technique in the two-dimensional case, it is convenient to write the discrete system to be solved in the following matrix form

$$\mathcal{D}_x\mathcal{U} + \mathcal{U}\mathcal{D}_y^T - \sigma\mathcal{U} = \mathcal{H}\,, \tag{3.157}$$

where $\mathcal{U}$ is the matrix of dimension $\overline{N}_x \times \overline{N}_y$ made with the inner unknowns, that is,

$$\mathcal{U} = [u_N(x_i, y_j)]\,,\quad i = 1,\ldots,\overline{N}_x\,,\quad j = 1,\ldots,\overline{N}_y\,.$$

In Eq.(3.157), $\mathcal{D}_x$ and $\mathcal{D}_y$ are the matrices of dimension $\overline{N}_x \times \overline{N}_x$ and $\overline{N}_y \times \overline{N}_y$, respectively, analogous to the $(N-1)\times(N-1)$ matrix $\mathcal{D}_x$

defined in the one-dimensional case considered in Section 3.7.1. Lastly, $\mathcal{H}$ is the $\overline{N}_x \times \overline{N}_y$ matrix containing the inner values of f and the values of g.

Let us denote by Λ_x and Λ_y the diagonal matrices whose entries are the eigenvalues $\lambda_{x,i}$, $i = 1, \ldots, \overline{N}_x$, and $\lambda_{y,j}$, $j = 1, \ldots, \overline{N}_y$, of the matrices $\mathcal{D}_x$ and $\mathcal{D}_y$, respectively, so that

$$\mathcal{D}_x = \mathcal{P}\Lambda_x\mathcal{P}^{-1}\,, \quad \mathcal{D}_y = \mathcal{Q}\Lambda_y\mathcal{Q}^{-1}\,, \tag{3.158}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are the matrices with their columns containing the eigenvectors.

Now the aim is to use the expressions (3.158) in order to reduce the system (3.157) to a sequence of uncoupled equations. First, Eq.(3.157) is left multiplied by $\mathcal{P}^{-1}$ so that it becomes

$$\mathcal{P}^{-1}\mathcal{D}_x\mathcal{P}\tilde{\mathcal{U}} + \tilde{\mathcal{U}}\mathcal{D}_y^T - \sigma\tilde{\mathcal{U}} = \tilde{\mathcal{H}}\,, \tag{3.159}$$

where $\tilde{\mathcal{U}} = \mathcal{P}^{-1}\mathcal{U}$ and $\tilde{\mathcal{H}} = \mathcal{P}^{-1}\mathcal{H}$. Taking the first equation (3.158) into account, the above equation gives

$$\Lambda_x\tilde{\mathcal{U}} + \tilde{\mathcal{U}}\mathcal{D}_y^T - \sigma\tilde{\mathcal{U}} = \tilde{H}\,. \tag{3.160}$$

Now we multiply at right this last equation by $(\mathcal{Q}^T)^{-1}$, that is,

$$\Lambda_x\hat{\mathcal{U}} + \hat{\mathcal{U}}\mathcal{Q}^T\mathcal{D}_y^T\left(\mathcal{Q}^T\right)^{-1} - \sigma\hat{\mathcal{U}} = \hat{\mathcal{H}}\,, \tag{3.161}$$

where we have introduced $\hat{\mathcal{U}} = \tilde{\mathcal{U}}\left(\mathcal{Q}^T\right)^{-1}$ and $\hat{\mathcal{H}} = \tilde{\mathcal{H}}\left(\mathcal{Q}^T\right)^{-1}$. Finally, taking into account that

$$\mathcal{Q}^T\mathcal{D}_y^T\left(\mathcal{Q}^T\right)^{-1} = \Lambda_y\,,$$

Equation (3.161) is of the form

$$\Lambda_x\hat{\mathcal{U}} + \hat{\mathcal{U}}\Lambda_y - \sigma\hat{\mathcal{U}} = \hat{\mathcal{H}}\,. \tag{3.162}$$

Therefore, if $\hat{\mathcal{U}} = [\hat{u}_{i,j}]$ and $\hat{\mathcal{H}} = \left[\hat{h}_{i,j}\right]$, $i = 1, \ldots, \overline{N}_x$, $j = 1, \ldots, \overline{N}_y$, Eq.(3.162) simply gives

$$\hat{u}_{i,j} = \frac{\hat{h}_{i,j}}{\lambda_{x,i} + \lambda_{y,j} - \sigma}\,, \quad i = 1, \ldots, \overline{N}_x\,, \quad j = 1, \ldots, \overline{N}_y\,. \tag{3.163}$$

Knowing $\hat{\mathcal{U}}$, it is easy to calculate $\tilde{\mathcal{U}}$ and $\mathcal{U}$.

To sum up, the algorithm is :

1. Calculate $\tilde{\mathcal{H}} = \mathcal{P}^{-1}\mathcal{H}$.
2. Calculate $\hat{\mathcal{H}} = \tilde{\mathcal{H}}\left(\mathcal{Q}^T\right)^{-1}$.
3. Calculate $\hat{\mathcal{U}}$ from (3.163).

4. Calculate $\tilde{\mathcal{U}} = \hat{\mathcal{U}}\mathcal{Q}^T$.
5. Calculate $\mathcal{U} = \mathcal{P}\tilde{\mathcal{U}}$.
6. Calculate the boundary values $u_N(x_i, y_j)$ for $(x_i, y_j) \in \Gamma_N^I$.

It must be noticed that the values of $u_N(x, y)$ at the corners of the square Ω are not at all involved in the solution. If necessary, they can be computed when the solution has been calculated. More precisely, let us consider, for example, the two sides $x = -1$ and $x = 1$ of Ω. The boundary condition (3.156) relative to these sides is extended up to the corners $(\pm 1, 1)$ to give the two equations

$$\alpha\, u_N\,(\pm 1, 1) + \beta\, \partial_x u_N\,(\pm 1, 1) = g\,(\pm 1, 1)\ .$$

Then, by expressing the derivatives $\partial_x u_N\,(\pm 1, 1)$ thanks to the formula (3.45) with $p = 1$, we get two equations that determine the two corner values $u_N\,(\pm 1, 1)$ since the inner values $u_N\,(x_i, 1)$, $i = 1, \ldots, \overline{N}_x$, are known.

In the case of the Poisson equation ($\sigma = 0$) with pure Neumann conditions ($\alpha = 0$), there exists a couple (i, j) for which $\lambda_{x,i} = \lambda_{y,j} = 0$ so that Eq.(3.163) has no meaning. As done in the one-dimensional case, the associated value $\hat{u}_{i,j}$ may be taken arbitrarily, say zero.

The computational effort associated with the matrix-diagonalization procedure is made of two parts. The first part consists of the calculation of the eigenvalues and eigenvectors, as well as the inversion of the eigenvector matrices. In the second part, described in the above algorithm, essentially four matrix-matrix products have to be performed.

The overall computational effort may seem to be very large and not really competitive with other solution methods (iterative methods, for example) for algebraic systems. As a matter of fact, in the case of the solution of unsteady problems (the usual situation considered here), Helmholtz problems similar to the one studied in the present section have to be solved at each time-cycle, that is a very large number of times. The interest of the method is that the first part of the calculations, which is also the most expensive, can be done once and for all in the preprocessing stage performed before the start of the time-integration. Therefore, at each time cycle, only the second part has to be performed. That is, the solution reduces to four matrix-matrix products which are efficiently done on vector computers.

Remarks

1. Change in boundary conditions : Influence matrix method

The matrix-diagonalization procedure described above works when the coefficients α and β in the boundary conditions (3.156) are constant on a whole side of the boundary Γ, although they may have different values on each side ; for example, the boundary conditions may be of Dirichlet type on three sides and of Neumann type on the fourth side. When this is not the case, that is, when α and β vary along one side, the algebraic system cannot be put in the form (3.157) and the procedure cannot be applied.

Such a situation appears, for example, when a Dirichlet condition is applied to one part of a side of Γ and a Neumann condition to the complementary part as considered in Section 8.4.1. In such a case, an efficient solution technique (Streett and Hussaini, 1986 ; Pulicani, 1988) based on the influence matrix method can be used.

The essence of this method, which can be applied to a large variety of problems as shown in the present book, is to take advantage of the linearity of the problem to construct the solution from a linear combination of elementary solutions. Then, the coefficients of this linear combination are determined so that the constructed solution satisfies the boundary conditions. One of the advantages of the method is that only Dirichlet problems have to be solved whatever the type of boundary conditions. The method is now described.

Let us consider the Helmholtz problem (3.155)-(3.156) in which α and β vary on Γ. The solution is sought in the form

$$u = \tilde{u} + \overline{u}\,, \tag{3.164}$$

where the part $\tilde{u}$ satisfies Eq.(3.155) with homogeneous Dirichlet boundary conditions, namely the $\tilde{\mathcal{P}}$-*Problem* :

$$\partial_{xx}\tilde{u} + \partial_{yy}\tilde{u} - \sigma\tilde{u} = f \ \text{ in } \Omega \tag{3.165}$$

$$\tilde{u} = 0 \ \text{ on } \Gamma\,. \tag{3.166}$$

The complementary part $\overline{u}$ is defined as the solution of the $\overline{\mathcal{P}}$-*Problem* :

$$\partial_{xx}\overline{u} + \partial_{yy}\overline{u} - \sigma\overline{u} = 0 \ \text{ in } \Omega \tag{3.167}$$

$$\overline{u} = \xi \ \text{ on } \Gamma\,, \tag{3.168}$$

where the unknown function ξ has to be determined such that $u = \tilde{u} + \overline{u}$ satisfies the boundary condition (3.156).

Now it is necessary to discuss the discretization of the problems with the Chebyshev collocation method. Let

$$u_N = \tilde{u}_N + \overline{u}_N \tag{3.169}$$

be the polynomial approximation associated with (3.164). The $\tilde{\mathcal{P}}$-problem is easily solved using the matrix-diagonalization procedure. The $\overline{\mathcal{P}}$-problem is solved by introducing the linear combination

$$\overline{u}_N(x,y) = \sum_{l=1}^{L} \xi_l \overline{u}_{N,l}(x,y)\,, \tag{3.170}$$

where L is the number of collocation points η_l on the boundary Γ_N^I, namely the discretized boundary without the corners. The elementary solution $\overline{u}_{N,l}$ satisfies the $\overline{\mathcal{P}}_l$-*Problem* :

$$\partial_{xx}\overline{u}_{N,l}(x_i,y_j) + \partial_{yy}\overline{u}_{N,l}(x_i,y_j) - \sigma\overline{u}_{N,l}(x_i,y_j) = f(x_i,y_j) \ \text{ for } (x_i,y_j) \in \Omega_N, \tag{3.171}$$

$$\overline{u}_{N,l|\eta_m} = \delta_{l,m} \quad \text{for } \eta_m \in \Gamma_N^I \,, \tag{3.172}$$

where $\delta_{l,m}$ is the Kronecker delta. Taking the boundary conditions (3.172) into account in (3.170), it is easily seen that the coefficients ξ_l, $l = 1, \ldots, L$, are nothing other than the values of the function ξ at the collocation points η_l belonging to Γ_N^I. Each $\overline{\mathcal{P}}_l$-problem is again solved by means of the diagonalization procedure. Then the constants ξ_l, $l = 1, \ldots, L$, are determined such that the boundary condition (3.156) is satisfied on Γ_N^I, that is,

$$\begin{aligned} &\alpha|_{\eta_m} \left(\tilde{u}_N|_{\eta_m} + \sum_{l=1}^{L} \xi_l \overline{u}_{N,l}|_{\eta_m} \right) \\ &\quad + \beta|_{\eta_m} \left(\partial_n \tilde{u}_N|_{\eta_m} + \sum_{l=1}^{L} \xi_l \partial_n \overline{u}_{N,l}|_{\eta_m} \right) = f|_{\eta_m} \,, \quad m = 1, \ldots, L \,, \end{aligned}$$

which can be written as

$$\mathcal{M}\Xi = \tilde{E} \,, \tag{3.173}$$

where $\Xi = (\xi_1, \ldots, \xi_L)^T$, $\mathcal{M} = [m_{i,j}]$, $i, j = 1, \ldots, N$, is the "influence matrix" defined by

$$m_{i,j} = \alpha|_{\eta_i} \, \overline{u}_{N,j}|_{\eta_i} + \beta|_{\eta_i} \, \partial_n \overline{u}_{N,j}|_{\eta_i}$$

and $\tilde{E}$ is the vector $\tilde{E} = [\tilde{e}_i]$, $i = 1, \ldots, L$, such that

$$\tilde{e}_i = f|_{\eta_i} - \alpha|_{\eta_i} \tilde{u}_N|_{\eta_i} - \beta|_{\eta_i} \partial_n \tilde{u}_N|_{\eta_i} \,.$$

Finally, the solution of (3.173) gives the values ξ_l, $l = 1, \ldots, L$, which enter into the linear combination (3.170), and the problem is completely solved.

The main part of the computational effort consists of the solution of $L + 1$ Dirichlet problems, the inversion of the matrix $\mathcal{M}$, and the linear combination of the elementary solutions. Again, the price may seemed to be very high if the problem has to be solved once only. However, as for the matrix-diagonalization procedure, the influence matrix method becomes very efficient when the problem has to be solved a large number of times in the course of an unsteady process. The interesting point is that, in such a case, the elementary solutions $\overline{u}_{N,l}$, $l = 1, \ldots, L$, do not depend on time. Therefore they can be calculated once and for all in the preprocessing stage. Also, the matrix $\mathcal{M}$ is inverted during this stage and its inverse $\mathcal{M}^{-1}$ is stored. Taking these precalculations into account, only one Dirichlet problem (for the determination of $\tilde{u}_N$) has to be solved, which amounts to four matrix-matrix products as seen above. Then the constants ξ_l, $l = 1, \ldots, L$, are determined from the product $\mathcal{M}^{-1}\tilde{H}$. Now it remains to evaluate the linear combination (3.170) at the collocation points $(x_i, y_j) \in \Omega_N \cup \Gamma_N^I$. The simple loop on the index l is costly. Computing time can be saved by reorganizing the evaluation as a matrix-vector product. Note that if

the storage of the L elementary solutions $\overline{u}_{N,l}(x_i, y_j)$ is not possible, the determination of the solution u_N can be done by solving the system

$$\partial_{xx} u_N(x_i, y_j) + \partial_{yy} u_N(x_i, y_j) - \sigma u_N(x_i, y_j) = f(x_i, y_j) \text{ for } (x_i, y_j) \in \Omega_N \,, \tag{3.174}$$

$$u_N|_{\eta_l} = \xi_l \text{ for } \eta_l \in \Gamma_N^I \,, \tag{3.175}$$

again with matrix diagonalization. It must be noticed that this last solution is a little more costly than the second one consisting of evaluating the linear combination by a matrix-vector product, but is much less expensive than the simple loop on l.

Finally, it is important to notice that the influence matrix technique is nothing other than a special algorithm for solving the full algebraic system obtained from the discretization of the equation (3.155) and the boundary conditions (3.156). This will be shown for the other applications of the influence matrix method in Sections 6.3.2 and 7.3.2.c.

2. Partial diagonalization procedure

In the diagonalization method described above, the operators in both directions are diagonalized (this method is sometimes called "full diagonalization") so that the system reduces to a set of uncoupled equations. However, it may be valuable, and sometimes necessary, to diagonalize in one direction only and solve the remaining system in the other direction.

For example, when solving the Helmholtz equation with the tau method, after diagonalization in one direction, the solution of the equation in the other direction is efficiently obtained by means of the technique leading to the solution of two uncoupled quasi-tridiagonal systems (see Section 3.4.1). We recall that such systems can be solved in a number of operations of the order of N, therefore, their solution is quite possible at each time-cycle of an unsteady process.

Another example is the case where one of the operators cannot be diagonalized. In such a case, the system corresponding to that direction can be solved by the direct inversion method, as explained in Section 3.4.2.

3.7.3 Three-dimensional equation

Now we consider the general three-dimensional problem for the Helmholtz equation :

$$\partial_{xx} u + \partial_{yy} u + \partial_{zz} u - \sigma u = f \text{ in } \Omega \tag{3.176}$$

$$\alpha u + \beta \partial_n u = g \text{ on } \Gamma = \partial\Omega \,, \tag{3.177}$$

where $\Omega = (-1, 1)^3$ and ∂_n represents the normal derivative to the boundary Γ. It is assumed that σ is a positive constant and that α and β keep a constant value on each side of the cube. The case where σ is variable has been mentioned in Remark 1 of Section 3.7.1. When α and β are variable,

nothing forbids, in principle, the use of the influence matrix method described in Remark 1 of Section 3.7.2. However, the method is not practical because of the size of the influence matrix. Therefore, for arbitrary functions α and β, the solution of the Helmholtz problem (3.176)-(3.177) must be obtained by techniques other than the matrix-diagonalization procedure, for example iterative methods. This is very much problem-dependent. For example, in the case where the boundary condition type change on the same side (Dirichlet on one part of the side and Neumann on the complementary part), the domain decomposition technique may be applied as mentioned in Section 8.4.1 in the two-dimensional case.

Problem (3.176)-(3.177) is solved by means of the collocation method based on the Gauss-Lobatto points

$$\begin{aligned} x_i &= \cos\frac{\pi i}{N_x}\,, \qquad i = 0,\ldots,N_x\,, \\ y_j &= \cos\frac{\pi j}{N_y}\,, \qquad j = 0,\ldots,N_y\,, \\ z_k &= \cos\frac{\pi k}{N_z}\,, \qquad k = 0,\ldots,N_z\,. \end{aligned} \tag{3.178}$$

As previously, we denote by Ω_N the open discretized domain, that is, $\Omega_N = \{x_i, y_j, z_k\}$, $i = 1,\ldots,\overline{N}_x = N_x - 1$, $j = 1,\ldots,\overline{N}_y = N_y - 1$, $k = 1,\ldots,\overline{N}_z = N_z - 1$, and by Γ_N^I the discretized boundary without the 12 edges of the cube.

The solution of the problem (3.176)-(3.177) is approximated with the polynomial $u_N(x,y,z)$ of degree at most equal to N_x, N_y, and N_z in x-, y- and z-directions, respectively.

The derivatives are approximated with the usual expressions [see Eq.(3.45)]. First, the Helmholtz equation (3.176) is forced to be satisfied by the polynomial $u_N(x,y,z)$ at every inner collocation point $(x_i,y_j,z_k) \in \Omega_N$. The boundary condition (3.177) is prescribed at every collocation point $(x_i,y_j,z_k) \in \Gamma_N^I$. Then, as previously, the boundary values $u_N(x_i,y_j,z_k)$ for $(x_i,y_j,z_k) \in \Gamma_N^I$ are eliminated thanks the boundary condition (3.177). We may write the resulting algebraic system under the tensor product form

$$\left(\mathcal{I}_z \otimes \mathcal{I}_y \otimes \mathcal{D}_x + \mathcal{I}_z \otimes \mathcal{D}_y \otimes \mathcal{I}_x + \mathcal{D}_z \otimes \mathcal{I}_y \otimes \mathcal{I}_x - \sigma \mathcal{I}_z \otimes \mathcal{I}_y \otimes \mathcal{I}_x\right) V = H\,, \tag{3.179}$$

where V is the column vector of (inner) unknowns ordered by row and horizontal section. More precisely, let us introduce the $\overline{N}_x$-component vector $V_{j,k}$:

$$V_{j,k} = \Big(u_N(x_1,y_j,z_k),\ldots,u_N(x_{\overline{N}_x},y_j,z_k)\Big)^T, \tag{3.180}$$

for $j = 1, \ldots, \overline{N}_y$, $k = 1, \ldots, \overline{N}_z$. Then we define the $\overline{N}_x \overline{N}_y$-component vector V_k by

$$V_k = \left(V_{1,k}, \ldots, V_{\overline{N}_y,k} \right)^T , \quad k = 1, \ldots, \overline{N}_z , \tag{3.181}$$

and, finally, V is the $\overline{N}_x \overline{N}_y \overline{N}_z$-component vector defined by

$$V = \left(V_1, \ldots, V_{\overline{N}_z} \right)^T . \tag{3.182}$$

In Eq.(3.179), $\mathcal{D}_x$, $\mathcal{D}_y$, and $\mathcal{D}_z$ are square matrices of dimensions $\overline{N}_x \times \overline{N}_x$, $\overline{N}_y \times \overline{N}_y$ and $\overline{N}_z \times \overline{N}_z$, respectively, analogous to the matrix $\mathcal{D}_x$ defined in Section 3.7.1 in the one-dimensional case ; $\mathcal{I}_x$, $\mathcal{I}_y$, and $\mathcal{I}_z$ are identity matrices in the x-, y- and z-directions having, respectively, the same dimensions as $\mathcal{D}_x$, $\mathcal{D}_y$, and $\mathcal{D}_z$. Lastly, H is the $\overline{N}_x \overline{N}_y \overline{N}_z$-component vector constructed in the same way as V, but with components $h_{i,j,k}$ calculated from the inner values of f and the values of g.

In Eq.(3.179), and in the following, the tensor product theory is used (see Halmos, 1958 ; Lynch *et al.*, 1964 ; Van Loan, 1992). Before discussing the details of the solution of (3.179) by the diagonalization process, we recall some definitions and properties of the tensor product (also called the "Kronecker product" or "direct product").

Let $\mathcal{A} = [a_{i,j}]$ and $\mathcal{B} = [b_{i,j}]$ be two (rectangular) matrices of dimensions $K \times L$ and $M \times N$, respectively. The tensor product $\mathcal{A} \otimes \mathcal{B}$ is the matrix of dimensions $KM \times LN$ defined by

$$\mathcal{A} \otimes \mathcal{B} = \begin{bmatrix} a_{1,1}\mathcal{B} & a_{1,2}\mathcal{B} & \ldots & a_{1,L}\mathcal{B} \\ a_{2,1}\mathcal{B} & a_{2,2}\mathcal{B} & \ldots & a_{2,L}\mathcal{B} \\ \ldots & \ldots & \ldots & \ldots \\ a_{K,1}\mathcal{B} & a_{K,2}\mathcal{B} & \ldots & a_{K,L}\mathcal{B} \end{bmatrix} . \tag{3.183}$$

The following relations are satisfied by tensor products

$$(\mathcal{A} + \mathcal{C}) \otimes \mathcal{B} = \mathcal{A} \otimes \mathcal{B} + \mathcal{C} \otimes \mathcal{B} , \tag{3.184}$$

$$(\mathcal{A} \otimes \mathcal{B}) (\mathcal{C} \otimes \mathcal{D}) = \mathcal{A}\mathcal{C} \otimes \mathcal{B}\mathcal{D} , \tag{3.185}$$

$$(\mathcal{A} \otimes \mathcal{B})^T = \mathcal{A}^T \otimes \mathcal{B}^T . \tag{3.186}$$

and in the case of invertible square matrices

$$(\mathcal{A} \otimes \mathcal{B})^{-1} = \mathcal{A}^{-1} \otimes \mathcal{B}^{-1} . \tag{3.187}$$

The first step in the solution of system (3.179) consists of exploiting the diagonalization process to put the solution V in a form involving only the inversion of "one-dimensional" matrices. First, system (3.179) is written as

$$\mathcal{A}V = H , \tag{3.188}$$

with

$$\mathcal{A} = \mathcal{I}_z \otimes \mathcal{I}_y \otimes \mathcal{D}_x^\sigma + \mathcal{I}_y \otimes \mathcal{D}_y \otimes \mathcal{I}_x + \mathcal{D}_z \otimes \mathcal{I}_y \otimes \mathcal{I}_x \,, \tag{3.189}$$

where $\mathcal{D}_x^\sigma = \mathcal{D}_x - \sigma\mathcal{I}_x$. The matrices $\mathcal{D}_x$, $\mathcal{D}_y$, and $\mathcal{D}_z$ are diagonalized according to

$$\mathcal{D}_x = \mathcal{P}\Lambda_x\mathcal{P}^{-1} \,, \;\; \mathcal{D}_y = \mathcal{Q}\Lambda_y\mathcal{Q}^{-1} \,, \;\; \mathcal{D}_z = \mathcal{R}\Lambda_z\mathcal{R}^{-1} \,, \tag{3.190}$$

where Λ_x, Λ_y, and Λ_z are the diagonal matrices whose entries are the eigenvalues of $\mathcal{D}_x$, $\mathcal{D}_y$ and $\mathcal{D}_z$, respectively. The matrices $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ are the associated eigenvector matrices.

Now, using (3.190) into (3.189), we have

$$\mathcal{R}^{-1} \otimes \mathcal{Q}^{-1} \otimes \mathcal{P}^{-1}\mathcal{A}\mathcal{R} \otimes \mathcal{Q} \otimes \mathcal{P} = \Lambda \,, \tag{3.191}$$

where

$$\Lambda = \mathcal{I}_z \otimes \mathcal{I}_y \otimes \Lambda_x^\sigma + \mathcal{I}_z \otimes \Lambda_y \otimes \mathcal{I}_x + \Lambda_x \otimes \mathcal{I}_y \otimes \mathcal{I}_x \,,$$

with $\Lambda_x^\sigma = \Lambda_x - \sigma\mathcal{I}_x$. The square matrix Λ of order $\overline{N}_x\overline{N}_y\overline{N}_z$ is diagonal, so that its inverse Λ^{-1} is easily calculated. Finally, from (3.191), we get

$$\mathcal{A}^{-1} = \mathcal{R} \otimes \mathcal{Q} \otimes \mathcal{P}\Lambda^{-1}\mathcal{R}^{-1} \otimes \mathcal{Q}^{-1} \otimes \mathcal{P}^{-1} \tag{3.192}$$

and

$$V = \mathcal{A}^{-1}H \,. \tag{3.193}$$

The second step of the algorithm is to calculate $\mathcal{A}^{-1}$ in the most efficient way. This amounts to calculating the vector

$$W = \mathcal{R}^{-1} \otimes \mathcal{Q}^{-1} \otimes \mathcal{P}^{-1}H \,, \tag{3.194}$$

then

$$\tilde{W} = \Lambda^{-1}W \,, \tag{3.195}$$

and, finally,

$$V = \mathcal{R} \otimes \mathcal{Q} \otimes \mathcal{P}\tilde{W} \,, \tag{3.196}$$

by using the same technique as for W.

Therefore, it remains to evaluate a product like W given by Eq.(3.194). Let us define the vectors X and Y :

$$X = \mathcal{I}_z \otimes \mathcal{I}_y \otimes \mathcal{P}^{-1}H \tag{3.197}$$

$$Y = \mathcal{I}_z \otimes \mathcal{Q}^{-1} \otimes \mathcal{I}_x X \,, \tag{3.198}$$

then

$$W = \mathcal{R}^{-1} \otimes \mathcal{I}_y \otimes \mathcal{I}_x Y \,. \tag{3.199}$$

Calculation of X. From the vector H we construct the rectangular matrix $\mathcal{H}$, of dimension $\overline{N}_x \times \overline{N}_y\overline{N}_z$, made of the juxtaposition of $\overline{N}_z$ rectangular matrices $\mathcal{H}_k$, $k = 1, \ldots, \overline{N}_z$, of dimension $\overline{N}_x \times \overline{N}_y$, let

$$\mathcal{H} = \left[\mathcal{H}_1 \;\; \mathcal{H}_2 \;\; \ldots \;\; \mathcal{H}_{\overline{N}_z}\right] .$$

The columns of each matrix $\mathcal{H}_k$ are made with the vector $H_{j,k}$, $j = 1, \ldots, \overline{N}_y$, defined similarly to Eq.(3.180), that is,

$$\mathcal{H}_k = \left[H_{1,k} \;\; H_{2,k} \;\; \ldots \;\; H_{\overline{N}_y,k}\right] , \;\; k = 1, \ldots, \overline{N}_z .$$

Then the matrix product

$$\mathcal{X} = \mathcal{P}^{-1}\mathcal{H}$$

defines the matrix $\mathcal{X}$ of dimension $\overline{N}_x \times \overline{N}_y\overline{N}_z$ whose columns are the $\overline{N}_y\overline{N}_z$ vectors $X_{1,1}, \ldots, X_{\overline{N}_y,1}, \ldots, X_{1,\overline{N}_z}, \ldots, X_{\overline{N}_y,\overline{N}_z}$ ordered in $\mathcal{X}$ in the same way that $H_{1,1}, \ldots, H_{\overline{N}_y,\overline{N}_z}$ were ordered in $\mathcal{H}$, more precisely,

$$\mathcal{X} = \left[\mathcal{X}_1 \;\; \mathcal{X}_2 \;\; \ldots \;\; \mathcal{X}_{\overline{N}_z}\right]$$

where the matrix $\mathcal{X}_k$ is

$$\mathcal{X}_k = \left[X_{1,k} \;\; X_{2,k} \;\; \ldots \;\; X_{\overline{N}_y,k}\right] , \;\; k = 1, \ldots, \overline{N}_z . \tag{3.200}$$

The sets of vectors $X_{1,k}, \ldots, X_{\overline{N}_y,k}$, $k = 1, \ldots, \overline{N}_z$, define the vector X by expressions similar to Eqs.(3.181) and (3.182), that is,

$$X_k = \left(X_{1,k}, \ldots, X_{\overline{N}_y,k}\right)^T , \quad k = 1, \ldots, \overline{N}_z ,$$

and

$$X = \left(X_1, \ldots, X_{\overline{N}_z}\right)^T . \tag{3.201}$$

Calculation of Y. This calculation makes use of the $\overline{N}_z$ matrices $\mathcal{X}_k$ previously calculated. Then the matrix products

$$\mathcal{Y}_k = \mathcal{X}_k \left(\mathcal{Q}^{-1}\right)^T , \;\; k = 1, \ldots, \overline{N}_z ,$$

define the matrices

$$\mathcal{Y}_k = \left[Y_{1,k} \;\; Y_{2,k} \;\; \ldots \;\; Y_{\overline{N}_y,k}\right] , \;\; k = 1, \ldots, \overline{N}_z , \tag{3.202}$$

of dimension $\overline{N}_x \times \overline{N}_y$ from which the vector Y is deduced in the same way as X.

Calculation of W. We construct the matrix $\mathcal{Z}$ of dimension $\overline{N}_x\overline{N}_y \times \overline{N}_z$:

$$\mathcal{Z} = \left[\mathcal{Y}_1^T \;\; \mathcal{Y}_2^T \;\; \ldots \;\; \mathcal{Y}_{\overline{N}_z}^T\right],$$

where $\mathcal{Y}_k^T$ is the transpose of the matrix $\mathcal{Y}_k$ calculated above. Then the product

$$\mathcal{W} = \mathcal{Z}\left(\mathcal{R}^{-1}\right)^T \tag{3.203}$$

defines the matrix $\mathcal{W}$ of dimension $\overline{N}_x\overline{N}_y \times \overline{N}_z$ written as

$$\mathcal{W} = \left[\mathcal{W}_1^T \;\; \mathcal{W}_2^T \;\; \ldots \;\; \mathcal{W}_{\overline{N}_z}^T\right]$$

with

$$\mathcal{W}_k = \left[W_{1,k} \;\; W_{2,k} \;\; \ldots \;\; W_{\overline{N}_y,k}\right]. \tag{3.204}$$

Finally, from the vectors $W_{j,k}$, we can construct the vector W, as done previously for X and Y, that is

$$W = \left(W_1, \ldots, W_{\overline{N}_z}\right)^T \tag{3.205}$$

where the vectors W_k, $k = 1, \ldots, \overline{N}_z$, are

$$W_k = \left(W_{1,k}, W_{2,k}, \ldots, W_{\overline{N}_y,k}\right)^T. \tag{3.206}$$

This ends the calculation of W. Then, as described above, the vector $\tilde{W}$ is easily calculated from Eq.(3.195) and, finally, the solution vector V, defined by (3.196), is calculated in the same way as W.

3.8 Iterative solution methods

Our opinion is that direct solution methods have to be recommended whenever possible, namely, when they are not too costly, i.e., concerning computing time or memory requirements. However, there exists a number of situations where the use of direct methods is not possible. The necessity to have recourse to iterative solution procedures appears for nonlinear or nonseparable linear equations. Also, an iterative solution is needed when the equation to be solved at each time-cycle of a time-marching procedure, although linear and separable, has time-dependent coefficients. Lastly, iterative procedures constitute an alternative to direct methods, especially in three-dimensional problems.

In this section, we briefly describe some usual iterative methods (see Eisenstat *et al.*, 1983 ; Canuto *et al.*, 1988 ; Quarteroni and Valli, 1994)

for the solution of linear algebraic systems arising from Chebyshev tau or collocation approximation methods, namely

$$\mathcal{A}\,V = F\,, \tag{3.207}$$

where V is the vector made with the unknowns. Efficiency requires preconditioning techniques, in which the preconditioning operator $\mathcal{A}_0$ has to be chosen according to the requirements given in Section 3.5.1.c.

The general algorithm is as follows :

1. Initialization

$$\begin{aligned} V^0 &\quad \text{is given} \\ R^0 &= \mathcal{A}\,V^0 - F \\ \mathcal{A}_0\,Y^0 &= R^0 \\ Z^0 &= Y^0\,, \end{aligned} \tag{3.208}$$

2. Current iteration

$$\begin{aligned} V^{m+1} &= V^m - \alpha_m\,Z^m \\ R^{m+1} &= R^m - \alpha_m\,\mathcal{A}\,Z^m \\ \mathcal{A}_0\,Y^{m+1} &= R^{m+1} \\ Z^{m+1} &= Y^{m+1} + \beta_m\,Z^m\,. \end{aligned} \tag{3.209}$$

The choice of the relaxation parameters α_m and β_m characterizes the type of procedure.

Preconditioned Richardson (PR)
This method is defined by

$$\alpha_m = \alpha = \text{ constant}\,, \quad \beta_m = 0\,. \tag{3.210}$$

The constant α has to be chosen to ensure convergence, that is, the spectral radius $\rho\,(\mathcal{E})$ of the iterative matrix $\mathcal{E} = \mathcal{I} - \alpha\,\mathcal{A}_0^{-1}\,\mathcal{A}$ is smaller than one. Assuming the eigenvalues λ of $\mathcal{A}_0^{-1}\,\mathcal{A}$ to be real positive, such that $\lambda_{min} \leq \lambda \leq \lambda_{max}$, the condition $\rho\,(\mathcal{E}) < 1$ is satisfied if

$$0 < \alpha < 2/\lambda_{max}\,. \tag{3.211}$$

The optimal value of α is $\alpha_{opt} = 2/\left(\lambda_{max} + \lambda_{min}\right)$ (see Isaacson and Keller, 1966).

Preconditioned Minimal Residual (PMR)
The technique is characterized by

$$\alpha_m = \frac{(R^m, \mathcal{A}\,Z^m)}{(\mathcal{A}\,Z^m, \mathcal{A}\,Z^m)}\,, \quad \beta_m = 0\,, \tag{3.212}$$

where $(\cdot,\cdot)$ is the Euclidian scalar product.

Preconditioned Conjugate Residual (PCR)
The parameters α_m and β_m are

$$\begin{aligned} \alpha_m &= \frac{(R^m, \mathcal{A}\, Z^m)}{(\mathcal{A}\, Z^m, \mathcal{A}\, Z^m)}\,, \\ \beta_m &= -\frac{\left(\mathcal{A}\, Y^{m+1}, \mathcal{A}\, Z^m\right)}{(\mathcal{A}\, Z^m, \mathcal{A}\, Z^m)}\,. \end{aligned} \tag{3.213}$$

Preconditioned Conjugate Gradient (PCG)
The parameters α_m and β_m are

$$\begin{aligned} \alpha_m &= \frac{(R^m, Y^m)}{(Z^m, \mathcal{A}\, Z^m)}\,, \\ \beta_m &= \frac{\left(R^{m+1}, Y^{m+1}\right)}{(R^m, Y^m)}\,. \end{aligned} \tag{3.214}$$

In general, PCR and PCG methods are designed for symmetric problems, but they have been found to work well for Chebyshev approximations, even if the associated matrices are not symmetric. Fröhlich and Peyret (1990) have compared the four iterative procedures described above in the case of the Stokes problem with variable density. The best convergence has been obtained with the PCR method. A similar conclusion was obtained for the solution of a steady advection-diffusion equation (Sabbah, 2000).

Among the iterative methods adapted to the solution of nonsymmetric systems, the Bi-CGSTAB (Van der Vorst, 1992) or its improved variant Bi-CGSTAB-QMR (Chan *et al.*, 1994) could be of interest. The Bi-CGSTAB(l) proposed by Sleijpen and Fokkema (1993) has been applied by Dimitropoulos and Beris (1997) to the spectral solution of the Helmholtz equation (3.155) with σ depending on x and y. We refer to Barret *et al.* (1994), Quarteroni and Valli (1994), or to Saad (1998) for a review of modern iterative techniques, efficient but which could be expensive when applied to systems derived from Chebyshev approximations.

http://www.springer.com/978-0-387-95221-5

Spectral Methods for Incompressible Viscous Flow

Peyret, R.

2002, XII, 434 p., Hardcover
ISBN: 978-0-387-95221-5