

# Spectral Methods

Computational Fluid Dynamics SG2212

Philipp Schlatter

Version 20100301

“Spectral methods” is a collective name for spatial discretisation methods that rely on an expansion of the flow solution as coefficients for ansatz functions. These ansatz functions usually have global support on the flow domain, and spatial derivatives are defined in terms of derivatives of these ansatz functions. The coefficients pertaining to the ansatz functions can be considered as a spectrum of the solution, which explains the name for the method.

Due to the global (or at least extended) nature of the ansatz functions, spectral methods are usually global methods, *i.e.* the value of a derivative at a certain point in space depends on the solution at all the other points in space, and not just the neighbouring grid points. Due to this fact, spectral methods usually have a very high order of approximation (*spectral convergence* meaning that the error with increasing resolution (number of grid points  $N$ ) is in fact decreasing exponentially ( $\propto (L/N)^N$ ) as opposed to algebraically ( $\propto (L/N)^p$ ) as for finite-difference methods). In addition, dispersion and diffusion properties of the derivative operator are advantageous compared to finite-difference methods. This can be easily seen by considering the modified wave number concept: Spectral methods usually give the *exact* derivative of a function, the only error is due to the truncation to a finite set of ansatz functions/coefficients.

On the other hand, spectral methods are geometrically less flexible than lower-order methods, and they are usually more complicated to implement. Additionally, the spectral representation of the solution is difficult to combine with sharp gradients, *e.g.* problems involving shocks and discontinuities. But for certain problems (mainly elliptic/parabolic problems in simple geometries) spectral methods are very adapted and efficient discretisation schemes.

In fact, spectral methods were among the first to be used in practical flow simulations. This was mainly prompted due to their high order of accuracy, meaning that an accurate solution could already be represented with a lower number of grid points. This cautious use of (expensive) computer memory was essential in the early days of CFD.

The topic of spectral methods is very large, and various methods and sub-methods have been proposed and are actively used. The following description aims at giving the fundamental ideas, focusing on the popular Chebyshev-collocation and Fourier-Galerkin methods.

# 1 Method of weighted residuals

## 1.1 Basic principle

Consider the initial-boundary-value problem of a partial differential equation

$$P[u] = 0 \quad (1)$$

on a domain  $\mathcal{D}$  for a function  $u(x, t)$  with boundary condition  $B(u) = 0$  on the boundary  $\partial B$  and initial condition  $u(x, 0) = u^0(x)$  at time  $t = 0$ . An ansatz for the approximate solution  $u_N(x, t)$  is made as a finite sum of known functions

$$u_N(x, t) = u_B(x, t) + \sum_{k=0}^N a_k(t) \cdot \phi_k(x) . \quad (2)$$

Here, the  $\phi_k(x)$  are called *trial functions* (ansatz functions) which are not changing with time, and  $a_k(t)$  are the corresponding time-dependent coefficients. Note that usually the  $\phi_k(x)$  fulfil homogeneous boundary conditions on  $\partial B$ , and the particular solution  $u_B(x, t)$  is used to satisfy the (possibly time-dependent) inhomogeneous boundary conditions.

The advantage of the ansatz (2) is that the temporal and spatial dependence and thus the partial derivatives are uncoupled. Therefore, spatial derivatives can be written as (assuming  $u_B = 0$ )

$$\frac{\partial^p u_N}{\partial x^p} = \sum_{k=0}^N a_k(t) \cdot \frac{d^p}{dx^p} \phi_k(x) = \sum_{k=0}^{N'} a_k^{(p)}(t) \cdot \phi_k(x) . \quad (3)$$

Note that depending on the  $\phi_k$  (e.g. for polynomials),  $N$  and  $N'$  might be different.

On inserting the series expression (2) into the original PDE (1), the residual is defined as

$$R(x, t) := P(u_N(x, t)) . \quad (4)$$

To determine the  $N + 1$  unknown coefficients  $a_k(t)$ , the method of weighted residuals requires that the residual  $R(x, t)$  multiplied with  $N + 1$  test function  $w_j(x)$  and integrated over the domain should vanish,

$$\int_{\mathcal{D}} w_j(x) \cdot R(x, t) dx = 0 , \quad j = 0, \dots, N , \quad (5)$$

or written using the scalar product  $(f, g) \equiv \int_{\mathcal{D}} f \cdot g dx$

$$(R, w_j) = 0 , \quad j = 0, \dots, N . \quad (6)$$

This means that the residual  $R$  is required to be orthogonal to all test functions (weights)  $w_j$ . This is the reason why the method is called *method of weighted residuals*.

## 1.2 Choice of test functions

There exist various methods to choose the test functions. Here, we only mention the two most common approaches, namely the Galerkin and the collocation method. Other important classes would be the Petrov-Galerkin method and the tau method.

**Galerkin method** For the Galerkin method (Boris Galerkin 1871–1845) the ansatz functions  $\phi_k(x)$  in equation (2) are chosen to be the same as the trial functions  $w_j(x)$ ,

$$w_j = \phi_j , \quad j = 0, \dots, N . \quad (7)$$

**Collocation method** A set of  $N + 1$  collocation points is chosen in the domain  $\mathcal{D}$  on which the residual  $R$  is required to vanish,

$$R(x_j) = 0 , \quad j = 0, \dots, N . \quad (8)$$

The consequence of this expression is that the original PDE (1) is fulfilled exactly in the collocation points,  $P(u_N)|_{x=x_j} = 0$ . Thus, the test functions become

$$w_j = \delta(x - x_j) , \quad j = 0, \dots, N , \quad (9)$$

with  $\delta$  being the Dirac delta function

$$\delta(x) = \begin{cases} 1 & \text{for } x = 0 \\ 0 & \text{otherwise .} \end{cases} \quad (10)$$

### 1.3 Choice of trial functions

The trial functions are usually smooth functions which are supported in the complete domain  $\mathcal{D}$ . There are many choices possible, in particular trigonometric (Fourier) functions, Chebyshev and Legendre polynomials, but also lower-order Lagrange polynomials with local support (finite element method) or b-splines. However, we focus on two important groups, the Fourier modes and Chebyshev polynomials.

#### 1.3.1 Fourier series

Fourier series are particularly suited for the discretisation of periodic functions  $u(x) = u(x + L)$  (Joseph Fourier 1768–1830). For such a periodic domain with periodicity  $L$ , we define the fundamental wave number  $\alpha = 2\pi/L$ , and the Fourier functions are

$$u_N(x) = \sum_{|k| \leq K} c_k e^{ik\alpha x} = \sum_{|k| \leq K} c_k \Phi_k , \quad \text{with } c_k \in \mathbb{C} . \quad (11)$$

The  $2K + 1 =$  coefficients  $c_k$  are the complex Fourier coefficients for the Fourier mode  $\Phi_k(x) = \exp(i\alpha kx)$ , see Figure 1. Note that the summation limits are sometimes also denoted as  $|k| \leq N/2$  with  $N = 2K$ . Additionally, a Fourier-transformed quantity is often denoted by a hat,  $\hat{u}_k = c_k$ .

A Fourier series of a smooth function (also in the derivatives, *i.e.* part of  $C_\infty$ ) converges rapidly with increasing  $N$ , since the magnitude of the coefficients  $|c_k|$  decreases exponentially. This behaviour is called *spectral convergence*. However, if the original function  $u(x)$  is non-continuous in at least one of the derivatives  $u^{(p)}(x)$ , the rate of convergence is severely decreased to order  $p$ , *i.e.*

$$\|u_N - u\| = \mathcal{O}(N^{-p}) , \quad N \rightarrow \infty , \quad (12)$$

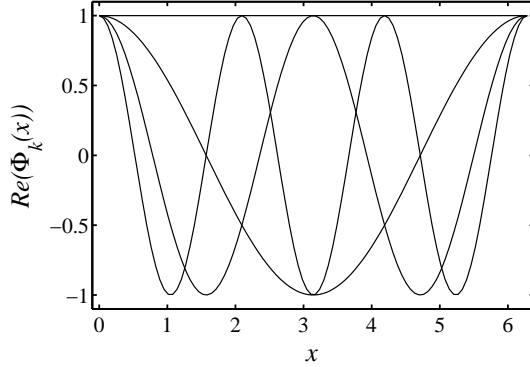


Figure 1: Fourier functions  $\Phi_k(x) = \exp(i\alpha kx)$  for  $k = 0, \dots, 3$  with  $\alpha = 2\pi/L = 1$ .

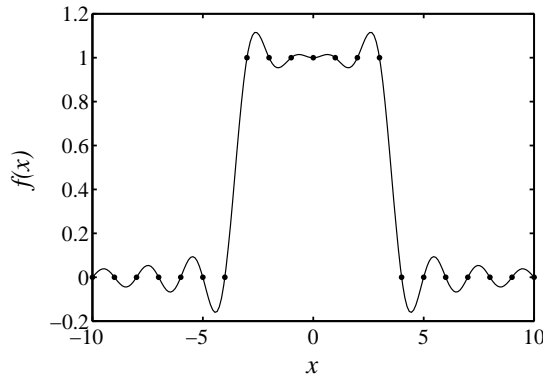


Figure 2: Illustration of the Gibbs phenomenon: Fourier interpolation of a function with sharp gradients leads to spurious oscillations near the discontinuity.

which corresponds to *algebraic convergence* as for finite-difference methods.

This phenomenon is usually referred to as the *Gibbs phenomenon* (Josiah Willard Gibbs 1839–1903), according to which spurious oscillations can be seen in the interpolant around sharp gradients. This behaviour is shown in Figure 2 for a step function featuring a discontinuity in the function itself, *i.e.* in  $u^{(0)}$ . Then, the order of convergence of the series expansion (2) is limited to zeroth order  $p = 0$ , *i.e.* the error does not decrease further with refinement of the grid.

The transformation from the space of the discrete representation of  $u_N$  (*physical space*) to the space of the Fourier components  $c_k$  (*spectral space*) is called the (forward) discrete Fourier transform  $\mathcal{F}(u_N)$ . Correspondingly, the reverse transform is the inverse Fourier transform  $\mathcal{F}^{-1}(c_k)$ . An efficient way to compute this is via the fast Fourier transform (FFT) (Cooley and Tukey 1965 going back to an idea by Carl Friedrich Gauss 1805), thereby reducing the computational effort from  $\mathcal{O}(N^2)$  to  $\mathcal{O}(N \log(N))$ . Note that there are various definitions of the scalings of the Fourier coefficients and the transforms.

Other important properties of Fourier series are:

- Orthogonality:

$$(\Phi_k, \Phi_l) = \frac{1}{L} \int_0^L \Phi_k(x) \Phi_l^*(x) dx = \frac{1}{L} \int_0^L \Phi_k(x) \Phi_{-l}(x) dx = \delta_{kl}, \quad (13)$$

with the Kronecker symbol  $\delta_{kl}$  and the complex conjugate  $\Phi_l^*(x)$ .

- Product rule:

$$\Phi_k \cdot \Phi_l = \Phi_{k+l} \quad (14)$$

- Differentiation:

$$\Phi_k'(x) = ik\alpha\Phi_k(x) \quad (15)$$

- Discrete transforms  $u_N(x_j) \leftrightarrow c_k$ : Based on the discrete orthogonality relation ( $N = 2K + 1$ ;  $|k|, |m| \leq K$ )

$$\sum_{j=1}^N e^{ikx_j} e^{imx_j} = \sum_{j=1}^N \Phi_k \Phi_m = N\delta_{k,-m} , \quad (16)$$

the relation between physical and spectral space is shown in a straight-forward way to be

$$u_N(x_j) = \sum_{k \leq |K|} c_k \Phi_k(x_j) , \quad c_k = N \sum_{j=1}^N u_N(x_j) \Phi_{-k} . \quad (17)$$

Note that  $n \in \mathbb{Z}$  corresponds to an arbitrary multiple of  $N$  which might lead to aliasing errors (see further down). These relations can be used to transform between the physical and spectral space, and are called a “discrete Fourier transform” (implemented *e.g.* using FFT).

### 1.3.2 Chebyshev polynomials

As stated, Fourier series are only a good choice for periodic function. For problems with non-periodic boundary conditions, ansatz functions based on orthogonal polynomials are preferred. One popular choice are the Chebyshev polynomials (Pafnutiy Lvovich Chebyshev, 1821–1894), defined on a domain  $x \leq |1|$  as

$$T_k(x) = \cos(k \arccos x) , \quad k = 0, 1, 2, \dots . \quad (18)$$

The first polynomials are thus (see also Figure 3)

$$T_0(x) = 1 \quad (19)$$

$$T_1(x) = x \quad (20)$$

$$T_2(x) = 2x^2 - 1 \quad (21)$$

$$T_3(x) = 4x^3 - 3x . \quad (22)$$

There exists also a recursion formula

$$T_{k+1}(x) + T_{k-1}(x) = 2xT_k(x) , \quad k \geq 1 . \quad (23)$$

A function  $u(x)$  is approximated via a finite series of Chebyshev polynomials as

$$u_N(x) = \sum_{k=0}^N a_k T_k(x) , \quad (24)$$

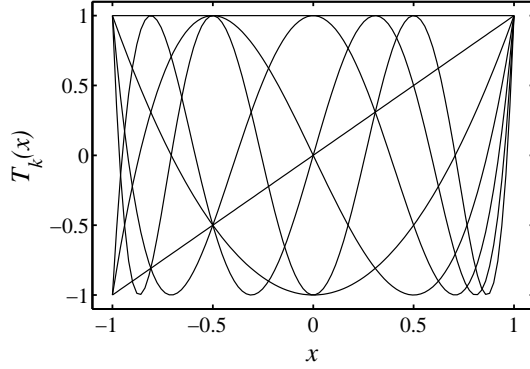


Figure 3: Chebyshev polynomials  $T_k(x)$  for  $k = 0, \dots, 6$ .

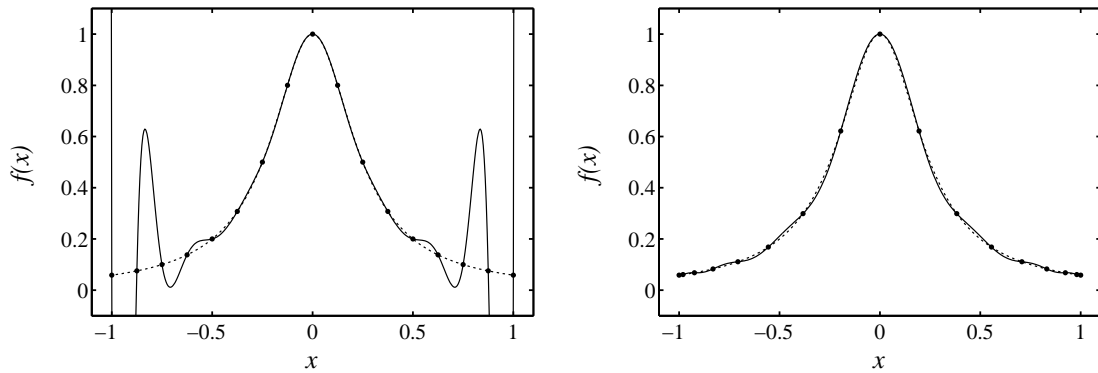


Figure 4: Illustration of the Runge phenomenon: Polynomial interpolation of the function  $f(x) = (1 + 16x^2)^{-1}$  on equidistant and non-equidistant (Gauss-Lobatto) grids.

with the  $a_k$  being the Chebyshev coefficients. Note that the sum goes from 0 to  $N$ , *i.e.* there are  $N + 1$  coefficients. The highest order of the polynomial approximation is thus  $N$ .

A general problem of high-order interpolation of a function via high-order polynomials is the *Runge phenomenon* (Carl David Tolm  Runge 1856–1927), similar to the Gibbs phenomenon for Fourier series. If a function is interpolated on an equidistant grid, the error grows as  $2^N$ . However, using a non-equidistant distribution of points such that the point density is (approximately) proportional to  $N\sqrt{1 - x^2}^{-1}$ , *i.e.* denser towards the domain boundaries, it can be shown that the interpolation errors decrease exponentially. A common distribution of points in particular for Chebyshev polynomials are the *Gauss-Lobatto* points

$$x_j = \cos \frac{\pi j}{N}, \quad j = 0, \dots, N. \quad (25)$$

The  $N + 1$  points  $x_j$  correspond to the locations of the extrema of  $T_N = \pm 1$ .

Chebyshev polynomials have a number of important properties:

- Alternating even and odd functions:

$$T_k(-x) = (-1)^k T_k(x). \quad (26)$$

- Orthogonality:

$$(T_k, T_l) = \int_{-1}^1 \frac{T_k(x)T_l(x)}{\sqrt{1-x^2}} dx = \frac{\pi}{2} c_k \delta_{kl} , \quad (27)$$

with  $c_0 = 2$ ,  $c_k = 1$  for  $k \geq 1$  and the Kronecker symbol  $\delta_{kl}$ .

- Boundary conditions:

$$T_k(1) = 1 \text{ and } T_k(-1) = (-1)^k . \quad (28)$$

- To evaluate the finite Chebyshev series (24) use the stable recursive algorithm

$$\begin{aligned} B_{N+1} &= 0 , \quad B_N = a_N \\ B_k &= a_k + 2xB_{k+1} - B_{k+2} , \quad k = N-1, \dots, 1 \\ u_N(x) &= a_0 - B_2 + B_1x . \end{aligned}$$

- The values of the Chebyshev polynomials on the Gauss-Lobatto nodes are

$$T_k(x_j) = \cos\left(\frac{kj\pi}{N}\right) , \quad j, k = 0, \dots, N$$

The transformation between the physical space  $u_N$  and spectral (Chebyshev) space  $a_k$  is done via the so-called Chebyshev transform. Since the Chebyshev polynomials are essentially cosine functions on a transformed coordinate, there exists a *fast* transform based on the FFT.

If a collocation method on the Gauss-Lobatto grid (25) is employed, the derivative of a discretised function  $u_N$  can be written as a matrix multiplication,

$$u'_N(x_i) = \sum_{j=0}^N D_{ij} u_N(x_j) . \quad (29)$$

The matrix  $\underline{\underline{D}} = [D_{ij}]$  is known as the (Chebyshev) *derivative matrix*, and represents a transformation into spectral space, spectral derivation and back-transform. An explicit form of the  $(N+1) \times (N+1)$ -matrix  $\underline{\underline{D}} = [D_{ij}]$  is given by

$$D_{ij} = \begin{cases} \frac{c_i (-1)^{i+j}}{c_j x_i - x_j} , & i \neq j \\ -\frac{x_i}{2(1-x_i^2)} , & 1 \leq i = j \leq N-1 \\ \frac{2N^2+1}{6} , & i = j = 0 \\ -\frac{2N^2+1}{6} , & i = j = N , \end{cases} \quad (30)$$

with

$$c_k = \begin{cases} 2 , & k = 0, N \\ 1 , & 1 \leq k \leq N-1 . \end{cases} \quad (31)$$

Note that the computation of  $\underline{\underline{D}}$  is very sensitive to round-off errors. Therefore, it is recommended for practical computations to use specifically adapted versions of the above formula, *e.g.* `chebdif.m` from “A Matlab Differentiation Matrix Suite” by J.A.C. Weideman and S.C. Reddy.

## 1.4 Example: Linear stationary case

Consider a linear problem of the form

$$P(u) \equiv Lu - r = 0 , \quad (32)$$

with the linear operator  $L$  and the inhomogeneous part  $r$  independent of  $u$ . The residual is then given by

$$R(x) = Lu_N - r . \quad (33)$$

Using the ansatz (2) in the equation (5) for the weighted residuals, one obtains a linear system of equations for the coefficients  $a_k$

$$\sum_{k=0}^N a_k \int_{\mathcal{D}} w_j \cdot L\phi_k(x) dx = \int_{\mathcal{D}} w_j (r - Lu_B) dx , \quad j = 0, \dots, N \quad (34)$$

or in matrix formulation  $\underline{\underline{A}}\underline{a} = \underline{s}$  with  $\underline{a} = [a_j]$  and

$$A_{jk} = \int_{\mathcal{D}} w_j \cdot L\phi_k(x) dx \quad s_j = \int_{\mathcal{D}} w_j \cdot (r - Lu_B) dx . \quad (35)$$

Depending on the choice of the test functions, the following cases can be derived

- Galerkin method:

$$A_{jk} = \int \phi_j L\phi_k dx , \quad s_j = \int \phi_j (r - Lu_B) dx \quad (36)$$

- Collocation method:

$$A_{jk} = L\phi_k(x_j) , \quad s_j = r(x_j) - Lu_B(x_j) \quad (37)$$

Consider for example the problem

$$u''(x) - H^2 \cdot u(x) = -1 , \quad |x| \leq 1 , \quad (38)$$

with the boundary conditions  $u(\pm 1) = 0$ . The exact solution is given as

$$u(x) = \frac{1}{H^2} \left( 1 - \frac{\cosh Hx}{\cosh H} \right) . \quad (39)$$

Then,  $L = \frac{d^2}{dx^2} - H^2$  and  $r = -1$ . Employing a collocation scheme, one gets

$$A_{jk} = \left( \frac{d^2}{dx^2} - H^2 \right) \phi_k(x) |_{x_j} \quad \text{and} \quad s_j = -1 . \quad (40)$$

The multiplication with  $a_k$  and summing according to  $\underline{\underline{A}}\underline{a} = \underline{s}$  gives

$$\sum_{k=0}^N a_k A_{jk} = \sum_{k=0}^N \left( \frac{d^2}{dx^2} - H^2 \right) a_k \phi_k(x) |_{x_j} = -1 . \quad (41)$$



The series ansatz of equation (2) now leads to

$$u_N''(x_j) - H^2 u_N(x_j) = -1, \quad j = 0, \dots, N, \quad (42)$$

which can be solved after choosing the trial functions and defining an appropriate derivative rule for the vector  $\underline{u}_N = [u_N(x_0), u_N(x_1), \dots]^T$ , cast in matrix form  $\underline{u}'_N = \underline{\underline{D}} \underline{u}_N$ . The algebraic system becomes with the identity matrix  $\underline{\underline{I}}$

$$(\underline{\underline{D}}^2 - H^2 \underline{\underline{I}}) \underline{u}_N = \underline{-1}, \quad (43)$$

which can be solved after implementation of the boundary conditions in terms of  $u_N$  (not  $a_k$ ). It can therefore be concluded that a collocation method relies on the description of the flow solution in physical space, replacing all derivatives with the corresponding derivative matrices  $\underline{\underline{D}}$ . The only reference to spectral space is via the definition of  $\underline{\underline{D}}$ . This is in contrast to the Galerkin method which we will present next.

## 1.5 Nonlinear problems

Consider a nonlinear (partial) differential equation  $P[u] = 0$ , *e.g.* the Burgers' equation (Johannes Martinus Burgers 1895–1981)

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, \quad 0 \leq x < 2\pi \quad (44)$$

with periodic boundary conditions on a domain  $L = 2\pi$ . Due to the periodicity, a Fourier Galerkin scheme shall be used for the spatial discretisation. The fundamental wave number  $\alpha = 2\pi/L = 1$ ; the trial functions are thus  $\phi_k = \Phi_k = e^{ikx}$  and the test functions  $\phi_l = \Phi_l = e^{ilx}$ . The approximation for the solution is

$$u_N(x, t) = \sum_{k=-K}^K \hat{u}_k(t) \phi_k(x) = \sum_{k=-K}^K \hat{u}_k(t) e^{ikx}. \quad (45)$$

Inserting  $u_N$  into the Burger equation (44) gives

$$\sum_{k=-K}^K \frac{d\hat{u}_k}{dt} \phi_k(x) + \sum_{k=-K}^K \hat{u}_k \phi_k(x) \sum_{m=-K}^K \hat{u}_m \frac{d\phi_m(x)}{dx} = 0. \quad (46)$$

The next step is to multiply with the  $N+1$  test functions  $\phi_l$  together with the integration over the domain obtaining a system of equations with  $l = 0, \dots, N$

$$\int_0^{2\pi} \phi_l(x) \sum_{k=-K}^K \frac{d\hat{u}_k}{dt} \phi_k(x) dx + \int_0^{2\pi} \phi_l(x) \sum_{k=-K}^K \hat{u}_k \phi_k(x) \sum_{m=-K}^K \hat{u}_m \frac{d\phi_m(x)}{dx} dx = 0, \quad (47)$$

which can be rewritten as follows using the product and derivative properties of the  $\phi_i$

$$\sum_{k=-K}^K \frac{d\hat{u}_k}{dt} \int_0^{2\pi} \phi_k(x) \phi_l(x) dx + \sum_{k=-K}^K \sum_{m=-K}^K im \hat{u}_k \hat{u}_m \int_0^{2\pi} \phi_l(x) \phi_{k+m}(x) dx = 0, \quad (48)$$

Employing the orthogonality relations (13)

$$\int_0^{2\pi} \phi_k(x)\phi_{-l}(x)dx = 2\pi\delta_{kl} \quad (49)$$

gives first

$$\sum_{k=-K}^K \frac{d\hat{u}_k}{dt} \delta_{-l,k} + \sum_{k=-K}^K \sum_{m=-K}^K im\hat{u}_k\hat{u}_m\delta_{-l,k+m} = 0, \quad (50)$$

which allows to get rid of one summation in each term yielding

$$\frac{d\hat{u}_{-l}}{dt} + \sum_{\substack{k=-K \\ -l=k+m \\ |m|\leq K}}^K im\hat{u}_k\hat{u}_m = 0, \quad (51)$$

and after replacing  $-l$  with  $l$  one obtains finally a system of nonlinear equations for the coefficients  $\hat{u}_l$

$$\frac{d\hat{u}_l}{dt} + \sum_{\substack{k=-K \\ l=k+m \\ |m|\leq K}}^K im\hat{u}_k\hat{u}_m = 0. \quad (52)$$

The fairly complex summation in the second term resembling a convolution is a consequence of the nonlinearity. Since we need to evaluate a sum for all components  $l$  of the solution, the number of operations is  $\mathcal{O}(K^2)$  which makes this sum evaluation the computationally most expensive part.

## 1.6 Pseudo-spectral method and aliasing errors

As we have seen, the evaluation of the nonlinear term in equation (52) pertaining to the Galerkin discretisation of the Burgers' equation is computationally of order  $\mathcal{O}(K^2)$ . It is therefore desirable to find more efficient ways to compute this sum. The most important way to do this is via Fourier transforms leading to an order  $\mathcal{O}(K \log K)$  for the same operation which is significantly less than  $\mathcal{O}(K^2)$  for large  $K$ .

To illustrate such an algorithm, consider the product  $w_j = w(x_j)$ ,  $j = 1, \dots, N$  of two grid functions  $u_j = u(x_j)$  and  $v_j = v(x_j)$  in physical space

$$w_j = u_j \cdot v_j. \quad (53)$$

We proceed by transforming to spectral space with the help of

$$u_j = \sum_{|k|\leq K} \hat{u}_k e^{ikx_j}, \quad \hat{u}_k = \frac{1}{N} \sum_{j=1}^N u_j e^{-ikx_j} \quad (54)$$

and the discrete form of the orthogonality relation

$$\frac{1}{N} \sum_{j=1}^N e^{ikx_j} e^{imx_j} = \delta_{k,-m+n \cdot N}. \quad (55)$$

In the last equation,  $n \in \mathbb{Z}$  corresponds to an arbitrary multiple of  $N$  which is closely related to aliasing errors (see further down). Finally one obtains

$$\hat{w}_l = \frac{1}{N} \sum_{|k| \leq K} \sum_{|m| \leq K} \hat{u}_k \hat{v}_m \sum_{j=1}^N e^{ikx_j} e^{i(m-l)x_j} = \sum_{\substack{k=-K \\ l=k+m \\ |m| \leq K}}^K \hat{u}_k \hat{v}_m , \quad (56)$$

*i.e.* the well-known result that a multiplication in physical space corresponds to a convolution in spectral space. To put it into other words, the fairly expensive evaluation of a convolution in spectral space, equation (56), is equivalent to the direct evaluation of a pointwise multiplication in physical space, equation (53).

Comparing our latest results, equation (56), to equation (52) we see that an efficient evaluation of a nonlinear product of two variables given in spectral space  $\hat{u}_k, \hat{v}_m = im\hat{u}_m$  is done via the following steps

1. Transform  $\hat{u}_k, \hat{v}_m$  to physical space using FFT:  $u_j = \mathcal{F}^{-1}(\hat{u}_k), v_j = \mathcal{F}^{-1}(\hat{v}_m)$
2. Multiplication in physical space:  $w_j = u_j \cdot v_j$
3. Transform back to spectral space:  $\hat{w}_l = \mathcal{F}(w_j)$ . This final results now reads:

$$\hat{w}_l = \sum_{\substack{k=-K \\ l=k+m+n \cdot N \\ |m| \leq K}}^K im\hat{u}_k \hat{u}_m . \quad (57)$$

Such an evaluation of the spectral convolution in physical space is usually termed *pseudo-spectral* evaluation of the nonlinear terms.

### 1.6.1 Aliasing errors

According to the sampling theorem, the highest wave number that can be represented as a grid function  $f_j$  with  $j = 1, \dots, N = 2K$  is  $K$ . Higher wave numbers  $k_h > K$  are mapped to this representable region  $|k| \leq K$  via

$$k = k_h - nN , \quad n \in \mathbb{Z} . \quad (58)$$

An example of such a case is given in Figure 5 illustrating the misrepresentation of wave numbers when mapped on a grid function with not enough grid points.

Such aliasing errors also appear when evaluating a convolution sum via Fourier transforms. To understand this, compare the nonlinear term in equations (52) and (57): The difference is the appearance of the term  $\dots + n \cdot N$  in the expression computed via the FFT. For the computed result to be correct, measures have to be taken to avoid the inclusion of these additional parts (*aliasing errors*) in the sum.

Another way to look at aliasing errors is by realising that the non-linear evaluation in  $\hat{w}_l$  in equation (56) involves factors of  $\hat{v}_m$  with  $|m| = |l - k| \leq 2K$ , *i.e.* wave numbers up to twice as high as representable on the grid. These wave numbers are thus mapped back to the represented wave-number space according to the above expression (58). Thus

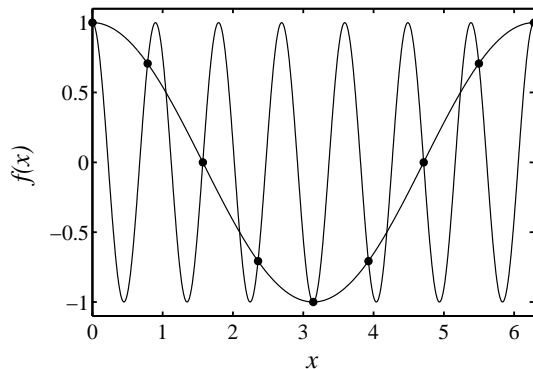


Figure 5: Illustration of aliasing errors, *i.e.* the mapping of higher wave numbers to lower ones. Here,  $K = 4$  and the wave numbers of the two waves are  $k_{1/2} = K \pm 3$ .

the sum will contain errors due to these spurious contributions, which are called *aliasing errors*.

These additional requirements when using a pseudo-spectral evaluation of the nonlinear terms are however not directly possible to implement using the above algorithm for computing  $\hat{w}_l$  given in equation (56). There are however some possibilities to remove (or at least reduce) aliasing errors. One popular variant is the so-called 3/2-rule: The original grid in physical space is refined by a factor  $M = 3/2N$  in every direction, and the nonlinear multiplications are then performed on this finer grid. Afterwards, the results product in transformed to spectral space, cutting away the wave numbers with  $|k| > K$ .

## 2 Acknowledgment and further reading

The present notes are based to some extent on the script “Berechnungsmethoden der Energie- und Verfahrenstechnik” by Leonhard Kleiser, ETH Zürich, Switzerland.

The financial support by the FLOW and KCSE Graduate School is gratefully acknowledged.

There are many good books about spectral methods. In particular, the following references are suggested for further reading:

- C. Canuto, M. Hussaini, A. Quarteroni and T. Zang. *Spectral Methods. Fundamentals in Single Domains*. Springer Verlag, 2006.
- C. Canuto, M. Hussaini, A. Quarteroni and T. Zang. *Spectral Methods. Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer Verlag, 2007.
- J.P. Boyd. *Chebyshev and Fourier Spectram Methods*. 2000.  
Online [http://www-personal.umich.edu/~jpboyd/BOOK\\_Spectral2000.html](http://www-personal.umich.edu/~jpboyd/BOOK_Spectral2000.html).
- L.N. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.
- J.A.C. Weideman and S.C. Reddy. A MATLAB differentiation matrix suite. *ACM Transactions of Mathematical Software*, Vol. 26, pp. 465-519, 2000.  
Download the code at <http://dip.sun.ac.za/~weideman/research/differ.html>.