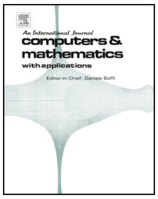
Computers and Mathematics with Applications 154 (2024) 175–198

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa



Approximate deconvolution discretisation

A. Boguslawski [a,*], A. Tyliszczak [a], B.J. Geurts [b,c]

[a] Czestochowa University of Technology, al. Armii Krajowej 21, 42-201 Czestochowa, Poland

[b] Mathematics of Multiscale Modeling and Simulation, Faculty EEMCS, University of Twente, PO Box 217, 7500 AE Enschede, the Netherlands

[c] Multiscale Physics of Energy Systems, Center for Computational Energy Research, Faculty of Applied Physics, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, the Netherlands

ARTICLE INFO

Keywords:
Finite differencing
Implied differencing filter
Approximate deconvolution
Computational fluid dynamics
Spectral accuracy

ABSTRACT

A new strategy is presented for the construction of high-order spatial discretisations extracted from a lower-order basic discretisation. The key consideration is that any spatial discretisation of a derivative of a solution can be expressed as the exact differentiation of a corresponding 'filtered' solution. Hence, each numerical discretisation method may be directly linked to a unique spatial filter, expressing the truncation error of the basic method. By approximately deconvolving the implied filter of the basic numerical discretisation an augmented high-order method can be obtained. In fact, adopting a deconvolution of the implied filter of suitable higher order enables the formulation of a new spatial discretisation method of correspondingly higher order. This construction is illustrated for finite difference (FD) discretisation schemes, solving partial differential equations in fluid mechanics. Knowing the implied filter of the basic discretisation, one can derive a corresponding higher order method by approximately eliminating the implied spatial filter to a certain desired order. We use deconvolution to compensate for the implied filter. This corresponds to a 'sharpening' of numerical solution features before the application of the basic FD method. The combination will be referred to as Approximate Deconvolution Discretisation (ADD). The accuracy of the deconvolved FD scheme depends on the order of approximation of the deconvolution filter. We present the 'sharpening' of several well-known FD operators for first- and second-order derivatives and quantify the achieved accuracy in terms of the modified wavenumber spectrum. Examples include high-order extensions up to new schemes with spectral accuracy. The practicality of the deconvolved FD schemes is illustrated in various ways: (i) by investigation of exactly solvable advection and diffusion problems, (ii) by tracking the evolution of the numerical solution to the Taylor-Green vortex problem and (iii) by showing that ADD yields spectral accuracy for the Burgers equation and for double-jet flow of an incompressible fluid.

1. Introduction

Numerical discretisation of derivatives in partial differential equations inherently introduces errors and hence a difference between the original formulation and the computational model that is achieved. As a result, numerical mathematics is concerned with the understanding, estimation and reduction of these errors. Different pathways can be distinguished. Adhering to structure-preserving discretisation in which important dynamic characteristics such as symmetries or conservation of key quantities is aimed at in the computational model [51,53], or, achieving high formal order of accuracy [2,5,38,45,54], are two important strategies.

The novelty in this paper derives from the basic observation that the truncation error associated with a particular spatial discretisation method can be expressed in terms of an induced convolution filter [15]. In turn, a low-order truncation error may be eliminated by appropriately deconvolving the corresponding induced filter. So, rather than 'accepting' that a basic discretization method comes with its truncation error, we investigate what benefits may be derived from an approximated deconvolution discretisation (ADD) approach. We discuss the theoretical basis, the technical implementation as well as the computational consequences and illustrate the ADD approach for a number of important canonical flow problems.

In this paper, we develop approximate deconvolution of the low-pass filter implied by a particular spatial discretisation of a derivative. This enables to compensate for the implied filter and achieve a higher-order method for the derivative. The implied filter is unavoidably connected to a chosen discretisation method, and will hence reduce the high-

[*] Corresponding author.
E-mail address: andrzej.boguslawski@pcz.pl (A. Boguslawski).

https://doi.org/10.1016/j.camwa.2023.11.039
Received 12 June 2023; Received in revised form 13 October 2023; Accepted 25 November 2023
Available online 5 December 2023

wavenumber content of a signal, i.e., 'blur' the original solution by filtering. However, knowing this filter explicitly, as is the case, e.g., for finite differencing methods, one may approximately invert it via deconvolution and combine this inverse with the basic discretisation method to 'sharpen' the composite derivative operator. This is referred to as Approximate Deconvolution Discretisation (ADD). Although most findings are general, we developed the approach with fluid-mechanical applications in mind. The main results in this paper are (i) the development of this framework, (ii) its application to the sharpening of basic finite differencing methods and (iii) the analysis of example flow predictions based on a particular sharpened method. The improvement of the formal order of accuracy is intimately connected to the accuracy with which the implied filter can be numerically approximated - it will be shown that even spectral-like accuracy may be achieved by appropriate deconvolution in periodic problems.

Inverse filtering is a general strategy that can be used to recreate some of the small scales in a given signal [12]. A well-known method is through deconvolution. Applications to image reconstruction and data science [7] enjoy growing interest. Also in the field of reduced order modelling, the coarsened description of a particular problem may induce new challenges to the computational modelling that may be mitigated using filter inversion [4]. In turbulent flow simulation, filter inversion has received ample attention for modelling of the sub-filter stress tensor in LES [13,35]. Also, models employing explicit, flow-dependent length-scale definitions received considerable attention recently [14,32,34]. Domaradzki and Saiki [9] developed a method to dynamically evaluate unresolved velocity fields based on kinematic adjustment of the resolved and unresolved fields. These approximately reconstructed fields could be included to evaluate the central closure problem in LES. Alternatively, Geurts [11] developed an approximate higher-order polynomial method for the direct inversion of the top-hat filter and proposed a generalized mixed similarity model. The inverse filtering of explicitly defined filters, in combination with the dynamic procedure was extensively tested on LES of the mixing layer by Kuerten et al. [23,65]. Stolz and Adams [42] proposed an alternative method of inversion which they called Approximate Deconvolution Model (ADM) based on repeated filtering according to the concept of Van Cittert [50] inversion. The ADM approach was applied by Aniszewski et al. [1] for modelling the subgrid surface tension in LES of two-phase flows.

Combustion in turbulent flow poses additional closure problems in which approximate inversion of the LES filter is a key step to reconstruct small-scale details of chemical species involved in the combustion. Wang and Ihme [46] and Wang et al. [47] applied inverse filtration for turbulent combustion modelling. They applied ADM in combination with Wiener inverse filtering [48]. Nikolaou and Vervisch [30] applied the ADM approach while Domingo and Vervisch [10] used the differential filter for inversion in turbulent combustion LES modelling. Three deconvolution methods were tested in the context of turbulent combustion modelling by Mehl et al. [29]: the approximate deconvolution method based on Van Cittert iterative deconvolution, a Taylor-based decomposition method, and the regularised deconvolution method based on minimisation of a quadratic cost criterion.

Another important use of filter inversion, which we follow in this paper, is to adopt it for the sharpening of a given spatial differentiation operator, thereby increasing the formal order of accuracy of that basic discretisation. Rather than accepting artificial dissipation and discretisation errors associated with a given numerical method as is pursued in Monotonically Integrated LES (MILES) [17] and quantified for channel flow in [22], we investigate the use of filter inversion to enhance the formal order of accuracy of the method. In fact, numerical differentiation based on central differencing can be interpreted as the exact derivative of the function filtered with the top-hat filter [49].

The implied filter characteristic of a given spatial discretisation was generalized by Geurts and Van der Bos [15], who showed that finite differencing of an arbitrary order implies a filter composed of a series of local top-hat filters. The idea behind ADD is the following. In fact, approximating the partial derivative of a function u in the point x_i, i.e., $\partial_x(u(x_i))$, using a finite difference method denoted by $\delta_x(u(x_i))$ then $\delta_x u(x_i) = \partial_x(L(u(x_i)))$, where L denotes the implied filter associated with the particular discretisation that was selected. The implied filter is related to the truncation error of the finite differencing method δ_x. Now, instead of taking this truncation error for granted, an alternative discretisation can be formulated in which δ_x is not applied directly to the function u, but to an inverse-filtered representation $L_N^{-1}(u)$. Here, L_N^{-1} denotes an N-th order approximate inverse [11] of L. Basically, this approximate removal of the implied filter corresponds to

$$D_x(u_i) \equiv \delta_x\left(L_N^{-1}(u(x_i))\right) = \partial_x\left(L(L_N^{-1}(u(x_i))\right) \approx \partial_x(u(x_i)) \tag{1}$$

Hence, to arrive at a higher-order method $D_x(u) \equiv \delta_x(L_N^{-1}(u))$ for the derivative, the primary task is to determine the implied filter of δ_x and to construct good approximations to its inverse. In the academic case in which the inverse of the implied filter is known exactly, the numerical discretisation applied to the inverse filtered or deconvolved solution would correspond exactly to the desired partial derivative. This hints at another use of the ADD approach, i.e., that of upgrading an existing 'legacy code' [25] by systematic insertion of the approximate inverse of the implied filter of the given numerical differentiation.

The numerically implied filter will negatively influence the treatment of marginally resolved scales in a solution. Such situations can occur when simulating high-Reynolds-number turbulent flow for which spatial resolution is easily compromised. In [15] the competition between the numerically implied filter of characteristic width equal to the mesh spacing h, and the explicit LES filter, of characteristic width Δ was shown to both influence the LES accuracy. The intricate interaction between discretisation and modelling errors was approached using the error-landscape framework [28] in which partial cancellation and exchange of numerical and subgrid dissipation [52] were main features. In this paper, we pursue a novel use of inversion of a filter as this could also improve the numerical discretisation by 'sharpening' the method. The present paper develops the improvement of classical finite differencing schemes through inverse filtering of the implied filter associated with the basic discretisation method. An alternative to improving the accuracy of existing schemes using deconvolution was proposed earlier for finite differencing and finite volume methods [20,40].

Addressing the truncation error of a discretization method, as pursued here, finds its context in the wider problem of predicting and controlling the overall accuracy of a method. In fact, various other sources of error, with their own interaction dynamics need to be reckoned with [59,62]. The total error dynamics associated with the discretization of a partial differential equation in space and time entails much more than truncation error alone [66,67]. Other aspects such as structure preservation [63], conservation principles, symmetry [64] and the interaction with modelling errors [61,60,58,34] make up the resulting computational dynamics that approximates the actual underlying problem. Improving a basic numerical method by explicitly accounting for the truncation error of that method can help to clarify the role of truncation error in the total computational dynamics. Moreover, it can yield an approach to handle so-called 'legacy codes' in which a given low-order implementation of a complex problem can be improved with modest interference with the underlying original code [25].

The organization of this paper is as follows. In Section 2 we present the foundation that underpins the sharpening of finite differencing methods. The filter implied by a specific numerical discretisation is considered in 3 and approximated inversion options are specified. The new schemes are analysed in terms of discretisation error and modified wavenumber. Simulation results obtained with the spectral-like enhanced FD method based on (i) central second order derivatives, (ii) Fourier integration of the implied filter and (iii) Wiener inversion for the deconvolution are discussed for the linear convection-diffusion equation, the Burgers equation, the Taylor-Green flow, and the double-jet problem in Section 4. Concluding remarks are collected in Section 5.

2. Sharpening finite differencing methods by inversion of the implied filter

In this section, we lay the foundation for the sharpening of a given spatial discretisation. We will establish that a numerical discretisation of a derivative is equivalent to applying the analytical derivative operator to a corresponding filtered solution [15]. This links a numerical discretisation to a particular implied filter. Through approximate inversion of this implied filter, the original (low-order) discretisation method can be converted into a higher-order method. The basic framework is presented in this section - examples of numerical implementations are discussed in section 3.

To illustrate the general framework for sharpening a numerical discretisation, we consider the simple second-order accurate central discretisation

$$\delta_x(u)(x) = \frac{u(x+h) - u(x-h)}{2h} \tag{2}$$

to approximate the derivative of the function u at the location x. Here, h is referred to as the grid spacing. Equivalently, this discrete approximation may be rewritten as

$$\delta_x(u)(x) = \partial_x\left(\int_{x-h}^{x+h} d\xi\, \frac{u(\xi)}{2h} \right) = \partial_x(L(u)(x)) \tag{3}$$

The equivalence between (2) and (3) may be directly inferred from the Fundamental Theorem of Calculus [56]. The implied filter L is identified as

$$L(u)(x) = \int_{x-h}^{x+h} d\xi\, \frac{u(\xi)}{2h} \tag{4}$$

Related to the implied filter L, we define an approximate inversion L_N^{-1} such that [11]

$$L \circ L_N^{-1}(x^k) = x^k \quad ; \quad k = 0, 1, ..., N \tag{5}$$

Various methods to construct the approximate inversion L_N^{-1} are available in literature, e.g., [11,42].

The approximate inversion operator can be combined with the discrete derivative δ_x to yield our 'sharpened' derivative approximation,

$$D_x(u) \equiv \delta_x(L_N^{-1}(u)) = \partial_x(L \circ L_N^{-1}(u)) \tag{6}$$

Here, use was made of (3). This example of sharpening the well-known central discretisation (2) will be extended to general finite differencing methods next, in which we also establish the formal order of accuracy of D_x in relation to the accuracy of the inversion of the implied filter.

We consider the problem of numerically approximating the first-order derivative of a function u in one spatial dimension for an arbitrary finite differencing method on a uniform grid. The numerical derivative $\delta_x u(x)$ may be written as

$$\delta_x(u)(x) = \frac{1}{h} \sum_{j=-n}^{m} a_j u(x + jh) \tag{7}$$

where, at this stage, we allow for a continuous dependence on x - the specification to a discrete grid will be made below. We adopt a general discretisation on $m + n + 1$ nodes, defining the stencil $[-n, m]$. The discretisation weights $\{a_j\}$ can be adapted to make the numerical derivative comply with certain design conditions, e.g., to achieve a desired formal order of accuracy. This formulation can also be extended to non-uniform grids - this is not pursued here as it would entail only technical complications that could distract from the main point of this section.

Any finite differencing method can be characterized by its implied filter L, analogous to (3). In fact, L can be related to the discretisation weights $\{a_j\}$ and expressed as a weighted average of skewed top-hat filters [15] of width h

$$L(u)(x) = \sum_{j=-n+1}^{m} b_j \left(\frac{1}{h} \int_{x+(j-1)h}^{x+jh} u(\eta) d\eta \right) \tag{8}$$

where

$$b_j = \sum_{i=j}^{m} a_i, \quad j = -n + 1, \dots, m \tag{9}$$

The implied filter L in (8) depends on the stencil $[-n, m]$ and the grid-spacing h of the finite differencing. We will not include this dependence explicitly in the notation of L but rather assume it implicitly, for notational convenience.

The formal order of accuracy of the sharpened numerical derivative D_x defined in (6), will be determined next. Applying an approximate inverse filter L_N^{-1} of order N, it can be shown that the formal order of accuracy of D_x is also of order N. The proof requires three steps and is based on Taylor expansion alone. First, consider a solution u to be sufficiently smooth, i.e., for every location x_0 we have

$$\begin{aligned} u(x) = {} & u(x_0) + \partial_x u(x_0)(x - x_0) + \frac{1}{2}\partial_{xx} u(x_0)(x - x_0)^2 + ... \\ & + \frac{1}{N!}\partial_x^N u(x_0)(x - x_0)^N \\ & + \frac{1}{(N+1)!}\partial_x^{N+1} u(\zeta(x, x_0))(x - x_0)^{N+1} \end{aligned} \tag{10}$$

with explicit remainder written in terms of a smooth function $\zeta(x, x_0)$ taking values between x and x_0. Here, the Lagrange form for the remainder was adopted - one may also incorporate the equivalent Cauchy form (more details on the formulation of the remainder term may be found in [57]). Correspondingly, using (5), we can take the second step and have

$$\begin{aligned} L \circ L_N^{-1}(u)(x) = {} & \sum_{k=0}^{N} \frac{1}{k!}\partial_x^k u(x_0)(x - x_0)^k + \\ & + \frac{1}{(N+1)!}\left(L \circ L_N^{-1}\right)\left\{\partial_x^{N+1} u(\zeta)(x - x_0)^{N+1}\right\} \\ = {} & u(x) \\ & + \frac{1}{(N+1)!}\left(L \circ L_N^{-1} - Id\right)\left\{\partial_x^{N+1} u(\zeta)(x - x_0)^{N+1}\right\} \end{aligned} \tag{11}$$

where Id denotes the identity operator. Here, a characteristic contribution appears in terms of the difference of $L \circ L_N^{-1}$ and the identity operator. If L_N^{-1} were an exact inverse of L, the remainder term would be exactly zero for any smooth function u. More practically, for N-th order filters designed according to (5), we verify $(L \circ L_N^{-1} - Id)p = 0$ for any polynomial p of order N.

Finally, we can take the third step and show that the expression in (11) implies that the sharpened higher-order method D_x defined above in terms of L_N^{-1} is also of order N. In fact,

$$\begin{aligned} D_x(u)(x) & \equiv \delta_x(L_N^{-1}(u)(x) = \partial_x(L \circ L_N^{-1}(u)(x)) \\ & = \partial_x u(x) + \frac{1}{(N+1)!}\left(L \circ L_N^{-1} - Id\right)\partial_x\left\{\partial_x^{N+1} u(\zeta)(x - x_0)^{N+1}\right\} \\ & = \partial_x u(x) + O((x - x_0)^N) \end{aligned} \tag{12}$$

where use was made of the fact that ∂_x commutes with $L \circ L_N^{-1} - Id$ as we work with a single fixed length-scale parameter h. This enables to explicitly identify the remainder term in (12) of order N as proposed above and completes the proof.

The above analysis shows that a higher-order method $D_x(u) \equiv \delta_x(L_N^{-1}(u))$ for the first derivative can be obtained using 'sharpening'. The main tasks in constructing such a method are to first determine the implied filter of δ_x and subsequently construct good approximations to its inverse. In literature, there are several methods that could be used to construct approximate filter inversions. Apart from the polynomial kernels as proposed in [11], also van Cittert deconvolution is directly applicable, as was pioneered for turbulence in [42]. Moreover, exact Wiener inversion [48] could be employed to create a high-order numerical point of reference. As an illustration, we investigate the application of van Cittert's approach next and quantify the quality of the sharpened discretisation in terms of the induced modified wavenumber.

The van Cittert method [50] to approximate the inverse of a filter is based on the formal equivalence

$$L^{-1} = (I - (I - L))^{-1} = \sum_{k=0}^{\infty} (I - L)^k \tag{13}$$

by analogy with geometric series. Inspired by this we define

$$L_N^{-1} = \sum_{k=0}^{N} (I - L)^k \tag{14}$$

Some low order examples are

$$N = 0: \quad L_0^{-1} = I \tag{15}$$

$$N = 1: \quad L_1^{-1} = 2I - L \tag{16}$$

$$N = 2: \quad L_2^{-1} = 3I - 3L + L^2 \tag{17}$$

These approximate inverses are readily verified to be normalized if L is, i.e., $L_N^{-1}(c) = c$ if $L(c) = c$ for any constant function c. Corresponding to these examples we infer $D_x u = \delta_x(L_N^{-1}(u))$ as

$$N = 0: \quad D_{x,0}(u) = \delta_x u \tag{18}$$

$$N = 1: \quad D_{x,1}(u) = 2\delta_x u - \delta_x(L(u)) \tag{19}$$

$$N = 2: \quad D_{x,2}(u) = 3\delta_x u - 3\delta_x(L(u)) + \delta_x(L(L(u))) \tag{20}$$

In case $N = 0$ there is no sharpening, but for $N \geq 1$ explicit changes to δ_x are induced by the sharpening.

The sharpening of the second-order accurate central discretisation of the first derivative (2) will next be illustrated for some low-order examples. The induced filter associated with (2) is the top-hat filter as mentioned in (4), which implies

$$L(x^k) \equiv \int_{x-h}^{x+h} \frac{\xi^k}{2h}\, d\xi = \frac{1}{2(k+1)h}\Big((x+h)^{k+1} - (x-h)^{k+1}\Big) \quad ; \quad k = 0, 1, 2, \ldots \tag{21}$$

We readily infer $L(1) = 1$, $L(x) = x$ and $L(x^2) = x^2 + h^2/3$ showing that the discretisation is exact for functions $u = 1$ and $u = x$. Equivalently, the implied filter L leaves these functions invariant. The first monomial that is not invariant under L is $u = x^2$. Using the $N = 2$ van Cittert approximate inverse L_2^{-1} as defined above, we may verify that $L \circ L_2^{-1}(x^k) = x^k$ for $k = 0, 1, 2$. This illustrates that the van Cittert method generates 2-nd order approximate inverses satisfying requirement (5) for $N = 2$.

Filters with higher values of N can also be constructed. Important for the development of an ADD method of N-th order is that the combination of the implied filter L with the approximate N-th order inverse filter L_N^{-1} is such that $L \circ L_N^{-1}(x^k) = x^k$ for $k = 0, 1, 2, \ldots, N$. The construction of an approximate inverse can be performed in various ways. Apart from the van Cittert method as outlined above, the paper by Geurts [11] presents the development of high-order filters of arbitrary order, using polynomial kernels. Further extensions to higher order filters in the context of computational aeroacoustics (CAA) may be found in earlier work of Bogey and Bailley [68].

The higher-order filtering of an arbitrary smooth function u can be developed on the basis of higher-order filtering of any x^k. Approximating u with its Taylor expansion enables higher-order filtering of any smooth function, i.e., $|u - L(u)| = O(h^N)$ for an N-th order filter L.

The approximate inverses generated with the van Cittert method can be used to define possible sharpened discretisations. The characteristics of these methods can be represented in terms of the associated modified wavenumber. To that end we investigate the discrete derivative of a single Fourier mode $u(x) = \sin(kx)$ as a function of kh, where k denotes the wavenumber of the mode. The focus on $\sin(kx)$ is not a restriction but rests on the fact that any smooth function u can be expanded in its Fourier series. Determining what the effect of a filter is on a single Fourier-mode is crucial for understanding how any general smooth function u would be altered when the filter is applied to it. This also implies that the analysis of a single Fourier mode at general k is relevant for non-periodic functions as well. In fact, for finite domains, a suitable (odd, even or otherwise) extension of the domain implies adherence to specific non-periodic boundary conditions. These correspond to a particular selection of wavenumbers from the general Fourier series, which are acted upon by the filter. We may derive analytically for $u(x) = \sin(kx)$

$$L(u)(x) = \int_{x-h}^{x+h} d\xi\, \frac{u(\xi)}{2h} = \frac{1}{2h}\int_{x-h}^{x+h} d\xi\, \sin(k\xi) = \Big(\frac{\sin(kh)}{kh}\Big)\sin(kx) \tag{22}$$

For convenience we write $G(kh) = \sin(kh)/(kh)$. Likewise,

$$\begin{aligned}\delta_x(\sin(kx)) &= \frac{1}{2h}\Big(\sin(k(x+h)) - \sin(k(x-h))\Big) = \Big(\frac{\sin(kh)}{h}\Big)\cos(kx)\\ &= G(kh)\{k\cos(kx)\}\end{aligned} \tag{23}$$

These two explicit results enable us to analyse the sharpened discretisations defined above. In fact,

$$D_{x,0}(\sin(kx)) = \delta_x(\sin(kx)) = \Big(\frac{\sin(kh)}{kh}\Big)\{k\cos(kx)\} \tag{24}$$

This establishes also the order of this discretisation since

$$D_{x,0}(\sin(kx)) = \Big(1 - \frac{1}{3!}(kh)^2 + \frac{1}{5!}(kh)^4 - \ldots\Big)\{k\cos(kx)\} \tag{25}$$

from which we infer second order accuracy, in view of the scaling of the error with $(kh)^2$. Slightly more involved, we obtain for the next order sharpening

$$\begin{aligned}D_{x,1}(\sin(kx)) &= \delta_x(L_1^{-1}(\sin(kx))) = \delta_x(2\sin(kx) - G(kh)\sin(kx))\\ &= 2\delta_x \sin(kx) - G(kh)\delta_x \sin(kx)\\ &= \Big(2 - G(kh)\Big)\Big(G(kh)\Big)\{k\cos(kx)\}\end{aligned} \tag{26}$$

Through cancellation of lower order terms, a higher-order accuracy is achieved, expressed by

$$D_{x,1}(\sin(kx)) = \Big(1 - \frac{1}{3!}(kh)^4 + (\frac{2}{3!5!} - \frac{1}{7!})(kh)^6 - \ldots\Big)\{k\cos(kx)\} \tag{27}$$

This establishes fourth-order accuracy. Likewise, $D_{x,2}$ can be proven to yield sixth-order accuracy.

This section considered the exact evaluation of the involved filters and their approximate inverses and showed that sharpening indeed may upgrade a lower-order discretisation into a method with high-order accurate numerical differentiation. In actual implementations, the involved filters are not directly available and one should consider particular approximate quadrature rules instead. In the next section, we investigate the numerical implementation in more detail and present several spatial filters and the actual discretisation sharpening that may be achieved.

3. Numerical discretisation filter and approximate deconvolution

The numerical formulation of the spatial filters implied by finite difference discretisations of first-order (Subsection 3.1) and second-order (Subsection 3.2) partial derivatives will be presented. Several quadrature rules can be selected to approximate the implied filter – here we investigate the trapezoidal and the Cavalieri-Simpson rule [33,16] as well as the Fourier series approach [6]. The corresponding approximate deconvolution of these implied filters will be quantified. Next to van Cittert's method as illustrated in the previous Section, also Wiener type inverse filtering is available for periodic problems [48]. In this section we will focus on the latter inversion method and discuss the achievable increase of the order of accuracy of the discretisation.

3.1. Approximation of first order derivative

The discrete representation of the implied filter determines the properties of the 'sharpened' high-order discretisation method D_x on a given grid. To evaluate the implied filter, the integral in (8) should be approximated. We start with the trapezoidal rule for the numerical integration, which yields the corresponding implied filter

$$L^{TR}(u)(x_i) = \sum_{j=-n+1}^{m} b_j \frac{u_{i+j-1} + u_{i+j}}{2} \tag{28}$$

The superscript 'TR' emphasizes that this discrete filter corresponds to trapezoidal quadrature. Rearranging the summation, one obtains

$$\begin{aligned} L^{TR}(u)(x_i) &= \frac{1}{2} b_{-n+1} u_{i-n} +! \sum_{j=-n+1}^{m-1} \frac{1}{2}(b_j + b_{j+1}) u_{i+j} + \frac{1}{2} b_m u_{i+m} \\ &\equiv \sum_{j=-n}^{m} G_j^{TR} u_{i+j} \end{aligned} \tag{29}$$

where the discrete filter kernel is obtained as

$$\begin{aligned} G_{-n}^{TR} &= \frac{1}{2} b_{-n+1} \\ G_j^{TR} &= \frac{1}{2}(b_j + b_{j+1}) \quad \text{for } j = -n+1, \dots, m-1 \\ G_m^{TR} &= \frac{1}{2} b_m \end{aligned} \tag{30}$$

Trapezoidal integration of the implied filter over a width $\Delta = (m+n)h$ yields a particular discrete representation of the filter kernel as in (29). This representation can be combined with different options for its approximate inversion. Here, we restrict to periodic problems in the domain $0 < x < 1$, for which Wiener inversion is particularly well suited. This method of approximate inversion proceeds in a few steps. First, we consider the Fourier series of the discrete filter kernel over the computational domain

$$\hat{G}_k^{TR} = \frac{1}{N} \sum_{i=0}^{N-1} G_i^{TR} e^{-2\pi \underline{i} k x_i}; \; k = -K, \dots, K \tag{31}$$

where $\underline{i} = \sqrt{-1}$ and $N = 2K+1$ is an odd number of mesh points. We will not explicitly denote the dependence of the Fourier coefficients on n and m that define the filter but rather assume this implicitly for notational convenience. The Wiener-inverse filter kernel in Fourier space is defined as

$$\hat{Q}_k^{TR} = \frac{1}{\hat{G}_k^{TR} N^2} \tag{32}$$

and the Wiener-inverse filter kernel in physical space is found from the inverse Fourier transform

$$Q_i^{TR} = \sum_{k=-K}^{K} \hat{Q}_k^{TR} e^{2\pi \underline{i} k x_k}; \; i = 0, \dots, N-1 \tag{33}$$

With these definitions recalled, we next specify approximate deconvolution discretisations associated with finite differencing methods. We quantify properties of the different methods by (i) specifying the actual discretisation error as function of the grid spacing in case of solutions consisting of a basic Fourier mode, and (ii) by determining the modified wavenumber associated with the sharpened method. It will be shown that the order with which the implied filter is approximated is crucial for the achieved sharpening - we investigate the trapezoidal rule, Simpson integration and the spectral method.

The approximate deconvolution discretisation for central finite differencing schemes can be elaborated in detail in case the implied filter is evaluated using the trapezoidal rule. These central schemes correspond to $m = n$, and have $a_0 = 0$ and $a_{-j} = -a_j$, $j = 1, \dots, n$. In addition, there is symmetry in $b_{-j+1} = b_j$ and $G_{-j}^{TR} = G_j^{TR}$. Moreover, the coefficients for a_j sum up to 0, while the coefficients for b_j and G_j^{TR} sum up to 1 for consistency and normalisation reasons. The schemes of 2^{nd}, 4^{th} and 6^{th} order can be specified using (9) and (30) as follows (for G_j^{TR} only the entries for $j \geq 0$ are listed as the above symmetries allow to read off the components for $j < 0$)

- Second order accuracy ($n = 1$):

$$\begin{aligned} &a_{-1} = -\tfrac{1}{2} \qquad a_1 = \tfrac{1}{2} \\ &b_0 = \tfrac{1}{2} \qquad b_1 = \tfrac{1}{2} \\ &G_0^{TR} = \tfrac{2}{4} \qquad G_1^{TR} = \tfrac{1}{4} \end{aligned}$$

- Fourth order accuracy ($n = 2$):

$$\begin{aligned} &a_{-2} = \tfrac{1}{12} \quad a_{-1} = -\tfrac{8}{12} \qquad a_1 = \tfrac{8}{12} \quad a_2 = -\tfrac{1}{12} \\ &b_{-1} = -\tfrac{1}{12} \qquad b_0 = \tfrac{7}{12} \qquad b_1 = \tfrac{7}{12} \quad b_2 = -\tfrac{1}{12} \\ &G_0^{TR} = \tfrac{14}{24} \quad G_1^{TR} = \tfrac{6}{24} \quad G_2^{TR} = -\tfrac{1}{24} \end{aligned}$$

- Sixth order accuracy ($n = 3$):

$$\begin{aligned} &a_{-3} = -\tfrac{1}{60} \quad a_{-2} = \tfrac{9}{60} \quad a_{-1} = -\tfrac{45}{60} \qquad a_1 = \tfrac{45}{60} \quad a_2 = -\tfrac{9}{60} \quad a_3 = \tfrac{1}{60} \\ &b_{-2} = \tfrac{1}{60} \quad b_{-1} = -\tfrac{8}{60} \qquad b_0 = \tfrac{37}{60} \qquad b_1 = \tfrac{37}{60} \quad b_2 = -\tfrac{8}{60} \quad b_3 = \tfrac{1}{60} \\ &G_0^{TR} = \tfrac{74}{120} \quad G_1^{TR} = \tfrac{29}{120} \quad G_2^{TR} = -\tfrac{7}{120} \quad G_3^{TR} = \tfrac{1}{120} \end{aligned}$$

Fig. 1 shows the discrete kernels of the implied filters of the 2^{nd}, 4^{th} and 6^{th} order central differencing schemes in which the implied filter is evaluated on the basis of trapezoidal integration. The corresponding Wiener inverses are also included, showing that the compact support of the implied filters is accompanied by non-local inverse filter kernels that take non-zero values across the entire computational domain.

3.1.1. Implied filter based on trapezoidal integration

An insightful way of characterising the possible sharpening of a basic numerical discretisation method is obtained by considering the error of differentiating a given smooth function on a grid with mesh spacing h. For this purpose we consider a basic Fourier mode

$$u(x) = \sin(2\pi x) + \cos(2\pi x) \tag{34}$$

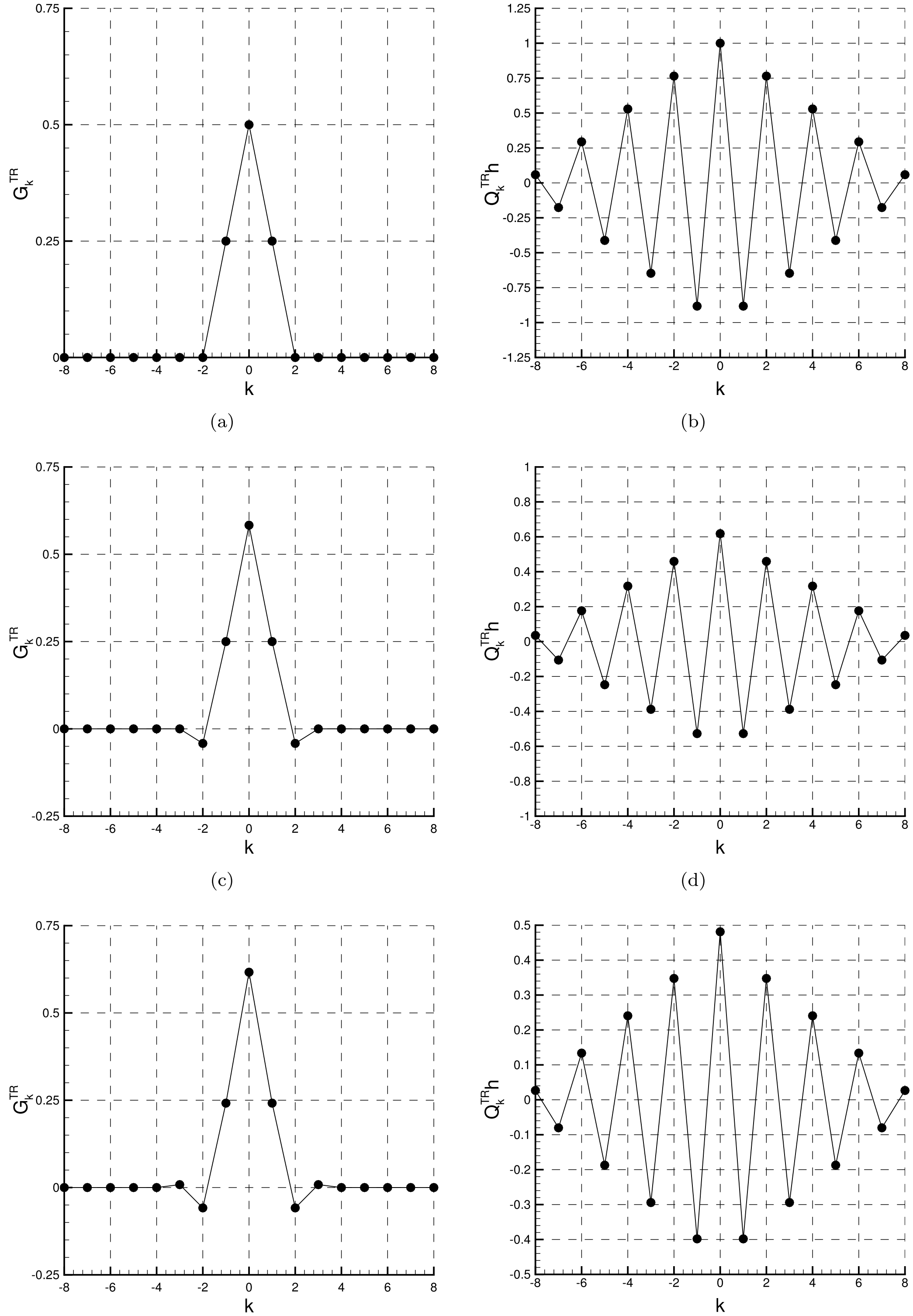
The discretisation error is defined as

$$\varepsilon^{TR} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \left[D_x^{TR}(u)(x_i) - \partial_x(u(x_i)) \right]^2 \right)^{1/2} \tag{35}$$

where D_x^{TR} denotes the 'sharpened' discrete differentiation method based on central differencing on a stencil with $2n+1$ nodes. The trapezoidal rule was used to approximate the implied filter, i.e.,

$$D_x^{TR}(u) \equiv \delta_x(L_W^{-1}(u)) \tag{36}$$

Here, L_W^{-1} is the Wiener evaluation to the inverse of L^{TR}. Following the previous Section, here one might also consider alternative, approximate inversions. However, since the Wiener inverse would be exact, this would imply the highest possible accuracy of the sharpened discretisation, thus creating a clear point of reference. This is why we adhere to the Wiener filter for the sake of simplicity.



(a)

(b)

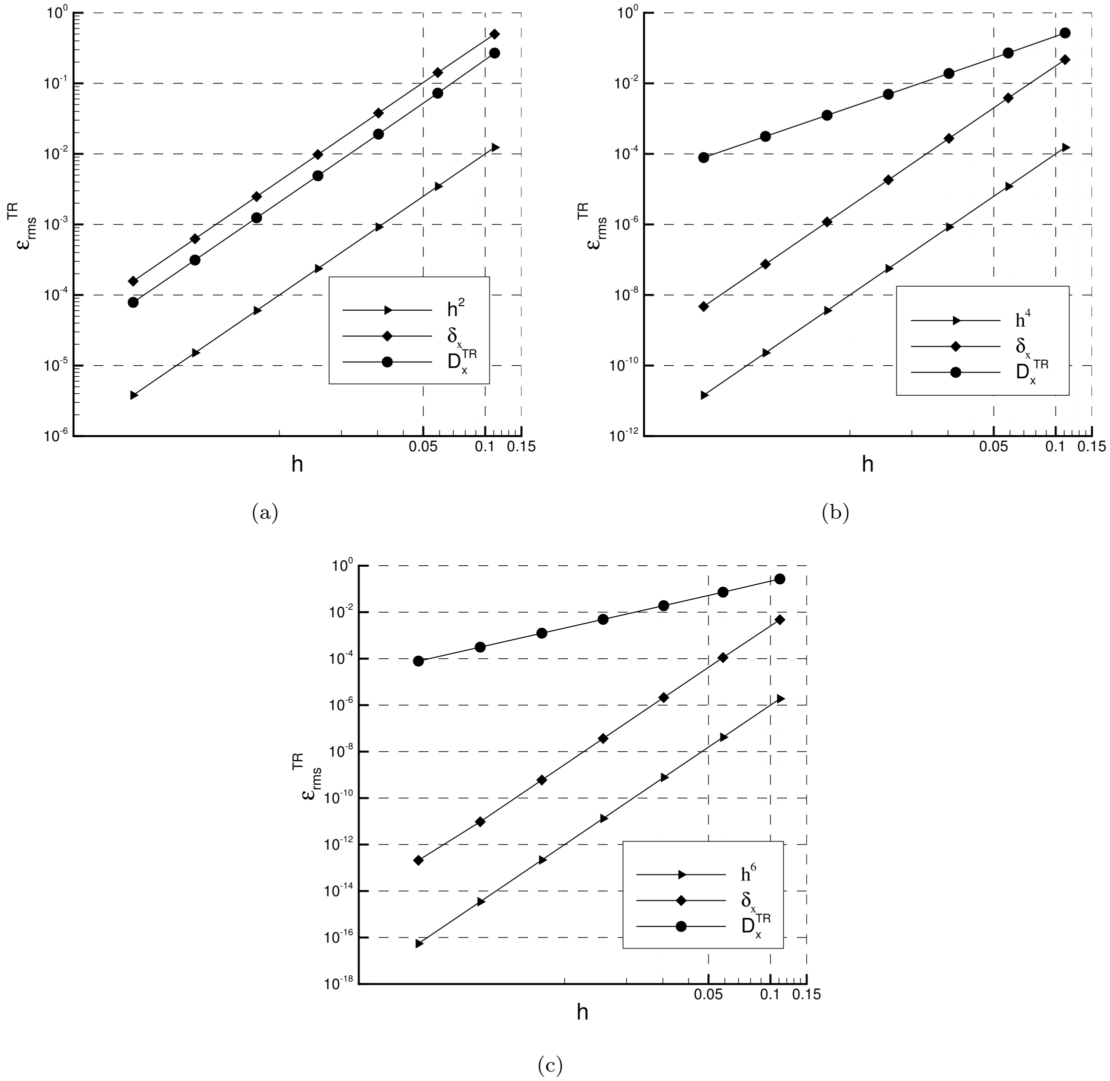
(c)

(d)

(e)

(f)

Fig. 1. The kernels for 2nd (a, b), 4th (c, d) and 6th (e, f) order central differencing methods with implied filter evaluated using trapezoidal integration. On the left the implied numerical filters are shown (a, c, e) and on the right the corresponding Wiener inverse filter (b, d, f).



(a)

(b)

(c)

Fig. 2. Error of differentiation of the first derivative using (a) 2nd, (b) 4th and (c) 6th order basic discretisation and trapezoidal quadrature to approximate the implied filter. The central discretisation δ_x on 3 (a), 5 (b) and (7) grid points and the sharpened discretisation D_x^{TR} are compared against the theoretical predictions.

The discretisation error in the first derivative of the basic Fourier mode as a function of grid spacing, obtained with 2^{nd}, 4^{th} and 6^{th} order central differencing, are shown in Fig. 2. It can be seen that for 2^{nd} order central differencing the 'sharpened' scheme (see Fig. 2a) slightly improves the accuracy compared to the case without trapezoidal sharpening. The total method D_x remains, however, of second order, i.e., similar to the basic unsharpened discretisation method δ_x. For 4^{th} and 6^{th} order central differencing, the use of trapezoidal integration in the evaluation of the implied filter does not yield any benefit. In fact, the complete 'sharpened' scheme remains second-order accurate irrespective of using the 4^{th} or 6^{th} order central differencing schemes as a point of departure. This illustrates that the order of accuracy of the 'sharpened scheme' is determined by the order of accuracy with which the implied filter is approximated and not by the order of accuracy of the basic discretisation method that is considered. In this instance, we work with the second-order accurate trapezoidal quadrature, which turns all 'sharpened' schemes second-order accurate as a consequence.

As a final characterisation of the sharpened schemes, we compute the modified wavenumber of the finite differencing schemes. A discrete function u can be represented via its Fourier coefficients on $2K+1=N$ grid points:

$$u(x)=\sum_{k=-K}^{K}\hat{u}_k e^{2\pi\underline{i}kx}, \quad x\in]0,1[\tag{37}$$

Following the notation proposed by Lele [26], it is convenient to introduce a scaled wavenumber w_k and scaled coordinate s as

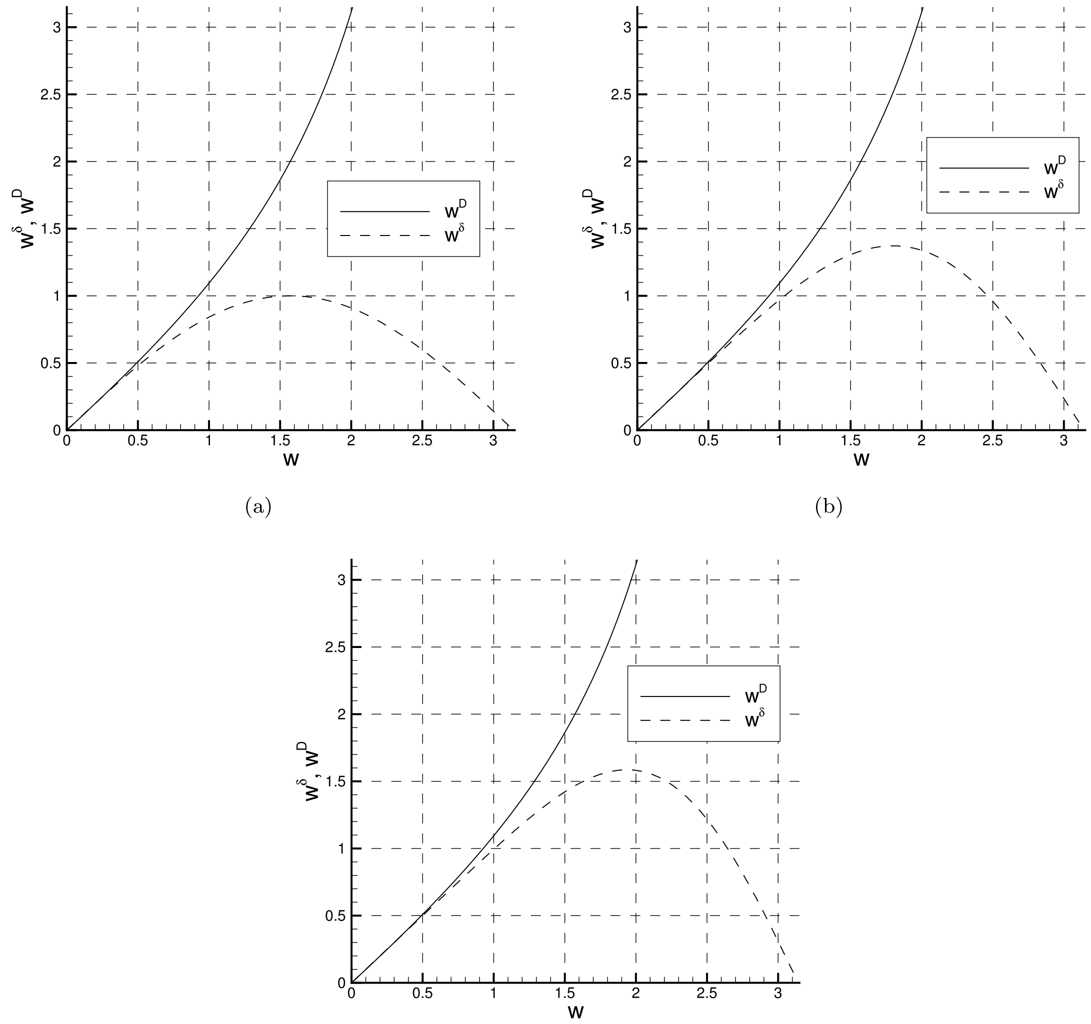
$$w_k=2\pi kh=\frac{2\pi k}{N}, \quad s=\frac{x}{h} \tag{38}$$

Using this notation the exact first derivative of u can be expressed as

$$\frac{du}{ds}=\sum_{k=-K}^{K}\underline{i}\hat{u}_k w_k e^{\underline{i}w_k s} \tag{39}$$

This can be compared to the corresponding expression resulting from the application of the numerical derivative. In fact, replacing the function u by its Fourier series and inserting this into (7), we may write for the central finite differencing scheme on $2n+1$ grid points

$$\delta_s(u(s_i))=\sum_{j=-n}^{n}\left(a_j\sum_{k=-K}^{K}\hat{u}_k e^{\underline{i}w_k s_{i+j}}\right)=\sum_{k=-K}^{K}\hat{u}_k\left(\sum_{j=-n}^{n}a_j e^{\underline{i}w_k s_j}\right)e^{\underline{i}w_k s_i} \tag{40}$$



(a)

(b)

(c)

Fig. 3. Modified wavenumber for (a) 2nd, (b) 4th, (c) 6th order finite differencing (dashed) and the corresponding sharpened scheme based on an implied filter defined in terms of trapezoidal integration (solid).

$$= \sum_{k=-K}^{K} \hat{u}_k \underline{i} \left(\sum_{j=-n}^{n} a_j \sin(w_k s_j) \right) e^{\underline{i} w_k s_i} \equiv \sum_{k=-K}^{K} \hat{u}_k \underline{i} W_k^\delta e^{\underline{i} w_k s_i}$$

where

$$W_k^\delta = \sum_{j=-n}^{n} a_j \sin\left(w_k s_j\right) \tag{41}$$

denotes the scaled wavenumber resulting from the numerical scheme. The latter wavenumber W_k^δ in general differs from the exact wavenumber corresponding to the first derivative - it is referred to as the 'modified' wavenumber and the difference compared to the exact value is a measure for the order of accuracy of the scheme and its truncation error, also at high wavenumbers.

Following the same steps, it can be shown that the sharpened approximate deconvolution schemes can be expressed as

$$D_x(u(s_i)) = \sum_{k=-K}^{K} \hat{u}_k \underline{i} W_k^D e^{\underline{i} w_k s_i} \tag{42}$$

where

$$W_k^D = W_k^\delta \left[\sum_{l=1}^{K} 2 Q_l^{TR} \cos\left(w_k l\right) + Q_0^{TR} \right] \tag{43}$$

Fourier errors for the three central differencing schemes considered are shown in Fig. 3. One can see that the modified wavenumbers as a function of the scaled wavenumber are nearly identical, no matter the original scheme that was adopted. In order to also achieve higher-order sharpened discretisation, we proceed with investigating higher-order quadrature of the implied filter and consider Cavalieri-Simpson integration next.

3.1.2. Implied filter based on Simpson integration

The implied filter L of any basic discretisation method can be approximated with different numerical quadrature methods. Obviously, this will influence the overall order of accuracy that can be achieved with the sharpening of that basic discretisation. In fact, with a perfect evaluation of the implied filter and its corresponding inversion L^{-1}, one may achieve analytically correct derivatives as $\partial_x u = \delta_x(L^{-1}u)$ corresponding to (3). The order of accuracy that the numerically approximated sharpening of the basic discretisation could yield equals the order of the quadrature method used to approximate the implied filter. The trapezoidal rule was shown to imply second-order accuracy of the sharpened methods, irrespective of the order of accuracy of the basic discretisation from which we started.

We elaborate on the characteristics of the sharpened methods that are obtained in case the implied filter is approximated using the higher-order Cavalieri-Simpson (CS) method [33]. For smooth solutions, fourth-order accuracy may be anticipated for the total sharpened discretisation. In the CS case, the implied filtration (8) can be approximated on the interval $[x_i - nh, x_i + mh]$ of a uniform grid as

$$
\begin{aligned}
L^{CS}(u)(x_i) &= \frac{1}{h} \sum_{\substack{j=-n+1 \\ j+n\ odd}}^{m-1} \left[b_j \int_{x_i+(j-1)h}^{x_i+jh} u(\eta)\, d\eta + b_{j+1} \int_{x_i+jh}^{x_i+(j+1)h} u(\eta)\, d\eta \right] \\
&= \sum_{\substack{j=-n+1 \\ j+n\ odd}}^{m-1} \left[b_j \left(\frac{5}{12} u_{i+j-1} + \frac{2}{3} u_{i+j} - \frac{1}{12} u_{i+j+1} \right) \right. \\
&\qquad \left. + b_{j+1} \left(-\frac{1}{12} u_{i+j-1} + \frac{2}{3} u_{i+j} + \frac{5}{12} u_{i+j+1} \right) \right]
\end{aligned}
\tag{44}
$$

Rearranging this expression yields

$$
\begin{aligned}
L^{CS}(u)(x_i) &= \left(\frac{5}{12} b_{-n+1} - \frac{1}{12} b_{-n+2} \right) u_{i-n} + \frac{2}{3} \left(b_{-n+1} + b_{-n+2} \right) u_{i-n+1} \\
&+ \sum_{\substack{j=-n+2 \\ j+n\ even}}^{m-2} \left[\left(\frac{5}{12} b_{j+1} - \frac{1}{12} b_{j+2} \right) + \left(-\frac{1}{12} b_{j-1} + \frac{5}{12} b_j \right) \right] u_{i+j} + \\
&+ \sum_{\substack{j=-n+2 \\ j+n\ odd}}^{m-2} \frac{2}{3} \left(b_j + b_{j+1} \right) \\
&+ \left[\frac{2}{3} \left(b_{m-1} + b_m \right) \Big|_{\substack{m+n \neq 2 \\ even}} + \left(-\frac{1}{12} b_{m-2} + \frac{5}{12} b_{m-1} \right) \Big|_{\substack{m+n \\ odd}} \right] u_{i+m-1} \\
&+ \left(-\frac{1}{12} b_{m-1} + \frac{5}{12} b_m \right) \Big|_{\substack{m+n \\ even}} u_{i+m}
\end{aligned}
\tag{45}
$$

From this, we may infer the discrete filter kernel for the quadratic interpolation as

$$
\begin{aligned}
G_{-n}^{CS} &= \frac{5}{12} b_{-n+1} - \frac{1}{12} b_{-n+2} \\
G_{-n+1}^{CS} &= \frac{2}{3} \left(b_{-n+1} + b_{-n+2} \right) \\
G_j^{CS} &= \begin{cases} -\frac{1}{12} b_{j-1} + \frac{5}{12} b_j \\ \quad + \frac{5}{12} b_{j+1} - \frac{1}{12} b_{j+2}, & j+n \text{ even} \\ \frac{2}{3} \left(b_j + b_{j+1} \right), & j+n \text{ odd} \end{cases} \qquad j = -n+2, \dots, m-2 \\
G_{m-1}^{CS} &= \begin{cases} \frac{2}{3} \left(b_{m-1} + b_m \right), & m+n \neq 2 \;\; m+n \text{ even} \\ -\frac{1}{12} b_{m-2} + \frac{5}{12} b_{m-1}, & m+n \text{ odd} \end{cases} \\
G_m^{CS} &= -\frac{1}{12} b_{m-1} + \frac{5}{12} b_m, \; m+n \text{ even}
\end{aligned}
\tag{46}
$$

In particular, discrete filter kernels for the central schemes for which $n = m$ are quantified as ($G_{-j}^{CS} = G_j^{CS}$):

$n = 1$

$$
G_{-1}^{CS} = \frac{1}{6}, \; G_0^{CS} = \frac{2}{3}, \; G_1^{CS} = \frac{1}{6}
\tag{47}
$$

$n = 2$

$$
G_{-2}^{CS} = -\frac{1}{12}, \; G_{-1}^{CS} = \frac{1}{3}, \; G_0^{CS} = \frac{1}{2}, \; G_1^{CS} = \frac{1}{3}, \; G_2^{CS} = -\frac{1}{12}
\tag{48}
$$

$n = 3$

$$
\begin{aligned}
&G_{-3}^{CS} = \frac{13}{720}, \; G_{-2}^{CS} = -\frac{7}{90} \; G_{-1}^{CS} = \frac{107}{720} \\
&G_0^{CS} = \frac{37}{45} \\
&G_1^{CS} = \frac{107}{720}, \; G_2^{CS} = -\frac{7}{90}, \; G_3^{CS} = \frac{13}{720}
\end{aligned}
\tag{49}
$$

Fig. 4 shows the kernels of the implied filters for 2^{nd}, 4^{th} and 6^{th} order basic central differencing schemes in case the fourth-order Cavalieri-Simpson integration is used to approximate the implied filter. Again, the Wiener method is used for exact inversion in the periodic setting. The discrete kernels of the inverse filters are seen to be qualitatively different from the corresponding kernels obtained with the trapezoidal integration (cf. Fig. 1). In fact, while the filter kernels themselves remain localized as seen previously, only the Wiener inverse kernel for $n = 2$ shows a non-zero value across the domain. In case $n = 1, 3$ the Wiener inverse kernel shows compact support instead.

The differentiation errors associated with (34) for the new 'sharpened' Cavalieri-Simpson schemes are shown in Fig. 5. The CS quadrature indeed yields effective 'sharpening' of the second-order central differencing method, yielding a total method of fourth order. This is the first example of improved accuracy obtained from sharpening. The CS method does not yield higher order for the $4th$ and $6th$ order basic methods - instead, overall fourth-order accuracy is observed, consistent with the expectations expressed above. Based on these examples, a clear framework for developing and analysing sharpened methods is obtained. This framework can be applied to higher-order central approaches as well as skewed upwind-biased discretisation. The latter is relevant for discretisation which adds numerical dissipation to the computational model. These directions are left for future studies. Here, we next investigate the maximum accuracy of sharpened schemes using all points in the computational domain for the approximate quadrature of the filter kernel.

3.1.3. Implied filter based on Fourier integration

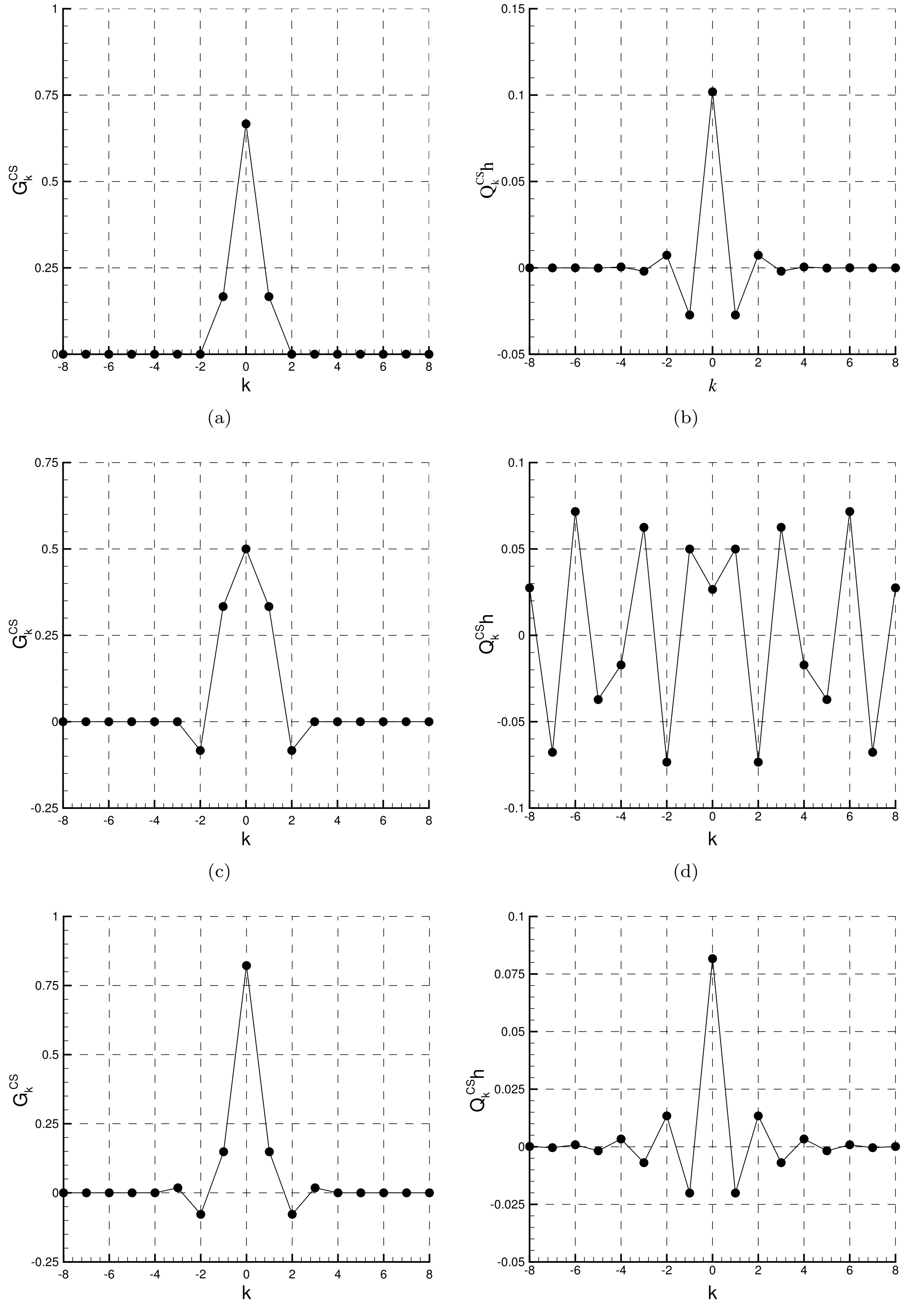
As considered here, the implied filtration can effectively be approximated by its Fourier series for periodic problems. In fact,

$$
\begin{aligned}
L^{Fo}(u)(x_i) &= \frac{1}{h} \sum_{j=-n+1}^{m} b_j \int_{x_i+(j-1)h}^{x_i+jh} u(\eta)\, d\eta \\
&\approx \frac{1}{h} \sum_{j=-n+1}^{m} b_j \int_{x_i+(j-1)h}^{x_i+jh} \Bigg(\sum_{\substack{k=-K \\ k \neq 0}}^{K} \widehat{u}_k e^{2\pi \underline{i} k \eta} + \widehat{u}_0 \Bigg) d\eta \\
&= \frac{1}{h} \sum_{j=-n+1}^{m} b_j \Bigg[\sum_{\substack{k=-K \\ k \neq 0}}^{K} \frac{\widehat{u}_k e^{2\pi \underline{i}(i+j)h} \left(1 - e^{-2\pi \underline{i} k h} \right)}{2\pi \underline{i} k} + \widehat{u}_0 h \Bigg]
\end{aligned}
\tag{50}
$$

where $u(\eta)$ was approximated over the interval $[x_i + (j-1)h, x_i + jh]$ by its Fourier series based on $(2K+1)$-terms. Introducing the inverse Fourier transform into the last equation leads to

$$
\begin{aligned}
L^{Fo}(u)(x_i) &= \frac{1}{h} \sum_{j=-n+1}^{m} b_j \Bigg[\sum_{\substack{k=-K \\ k \neq 0}}^{K} \frac{e^{2\pi \underline{i}(i+j)h} \left(1 - e^{-2\pi \underline{i} k h} \right)}{2\pi \underline{i} k N} \\
&\qquad \times \sum_{q=-K}^{K} u_{i+q} e^{-2\pi \underline{i} k (i+q) h} \\
&\qquad + \frac{h}{N} \sum_{q=-K}^{K} u_{i+q} \Bigg]
\end{aligned}
\tag{51}
$$

$$
= \sum_{q=-K}^{K} u_{i+q} \sum_{j=-n+1}^{m} b_j \Bigg[\sum_{\substack{k=-K \\ k \neq 0}}^{K} \frac{e^{2\pi \underline{i} k (j-q) h} \left(1 - e^{-2\pi \underline{i} k h} \right)}{2\pi \underline{i} k} + \frac{1}{N} \Bigg]
$$



(a)

(b)

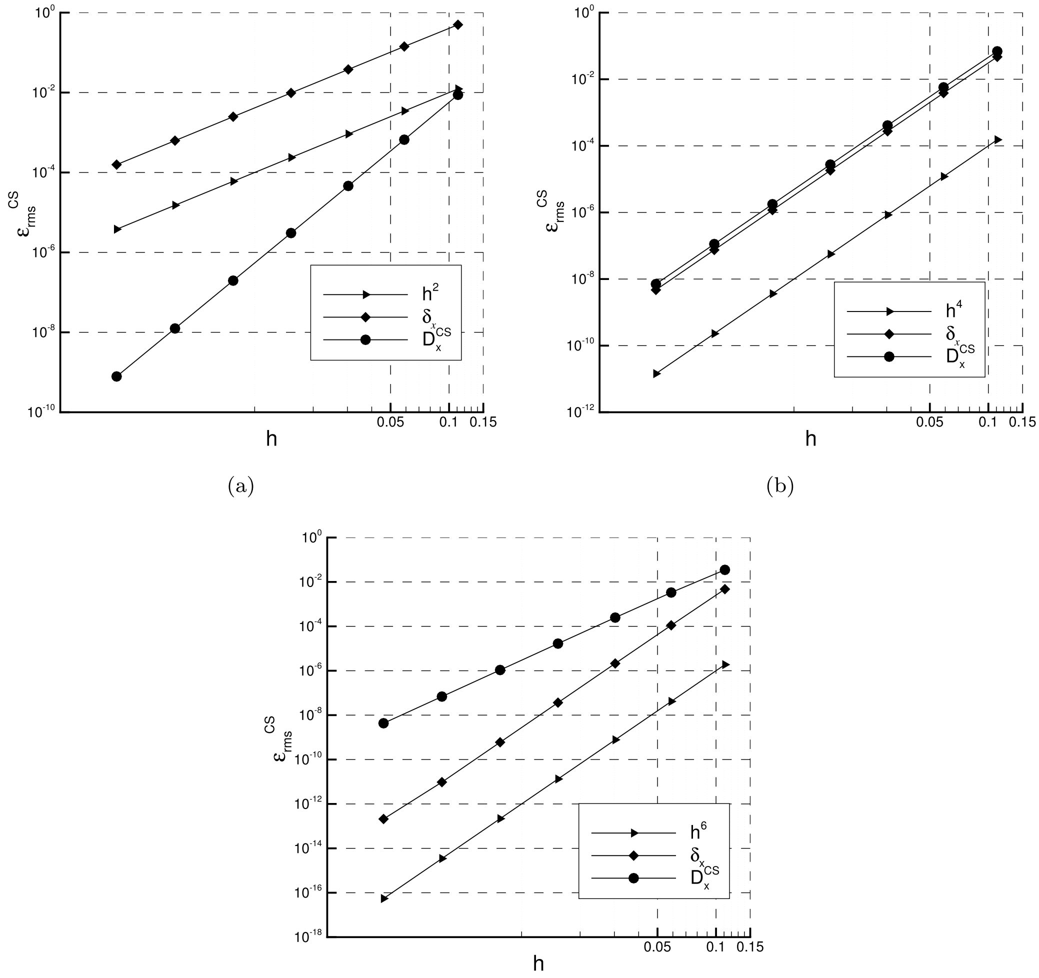
(c)

(d)

(e)

(f)

Fig. 4. The kernels for 2nd (a, b), 4th (c, d) and 6th (e, f) order central differencing methods with implied filter evaluated using Cavalieri-Simpson integration. On the left, the implied numerical filters are shown (a, c, e) and on the right the corresponding Wiener inverse filter (b, d, f).



(a)

(b)

(c)

Fig. 5. Error of differentiation of the first derivative using (a) 2nd, (b) 4th and (c) 6th order basic discretisation and Cavalieri-Simpon quadrature to approximate the implied filter. The central discretisation δ_x on 3 (a), 5 (b) and (7) grid points and the sharpened discretisation D_x^{CS} are compared against the theoretical predictions.

For the special case of central differencing where $n = m$, the discrete filter kernel associated with the Fourier approximation can be inferred as

$$G_q^{Fo} = \sum_{j=-n+1}^{n} b_j \Bigg\{ \sum_{k=1}^{K} \frac{1}{\pi k} \Big(\cos(2\pi k(j-q)h)\sin(2\pi kh) - \sin(2\pi k(j-q)h)(\cos(2\pi kh)-1) \Big) + \frac{1}{N} \Bigg\} \tag{52}$$

Fig. 6 shows the kernels of the implied filters associated with the 2^{nd}, 4^{th} and 6^{th} order central schemes and the kernels of the corresponding inverse filters, in case of Fourier integration. The differentiation errors arising from the smooth function (34) are shown in Fig. 7 for the 'Fourier-sharpened' central methods. Clearly, in all cases, spectral accuracy is obtained. The accuracy of all the sharpened schemes is verified to be independent of the order of the original method since Fourier integration yields results at machine accuracy. This is also readily verified from the modified wavenumber which corresponds up to machine accuracy to the exact result. This is shown in Fig. 8. The solid line refers to the sharpened schemes derived from the basic 2nd, 4th, and 6th order accurate finite differencing schemes. The use of the Fourier series interpolation of each of these basic finite differencing methods yields sharpened schemes with exact wave numbers for the first derivative. Consequently, the corresponding error dynamics of the sharpened scheme should be the same as in the case of the spectral scheme. It should be noted that if the Fourier series interpolation is replaced by a (high-degree) polynomial interpolation, a different method is generated, the characteristics of which will require global spectral analysis of the error dynamics. This study is subject of ongoing research, closely following the review in [55].

After having investigated the sharpening characteristics of different central differencing methods for approximating the first derivative, we next investigate the sharpening of the second derivative.

3.2. Approximation of second derivative

Very similar reasoning as presented above for the first derivative can be applied for 'sharpening' of the second derivative. To illustrate the approach, we consider an approximation of the second derivative using central differencing of second order accuracy

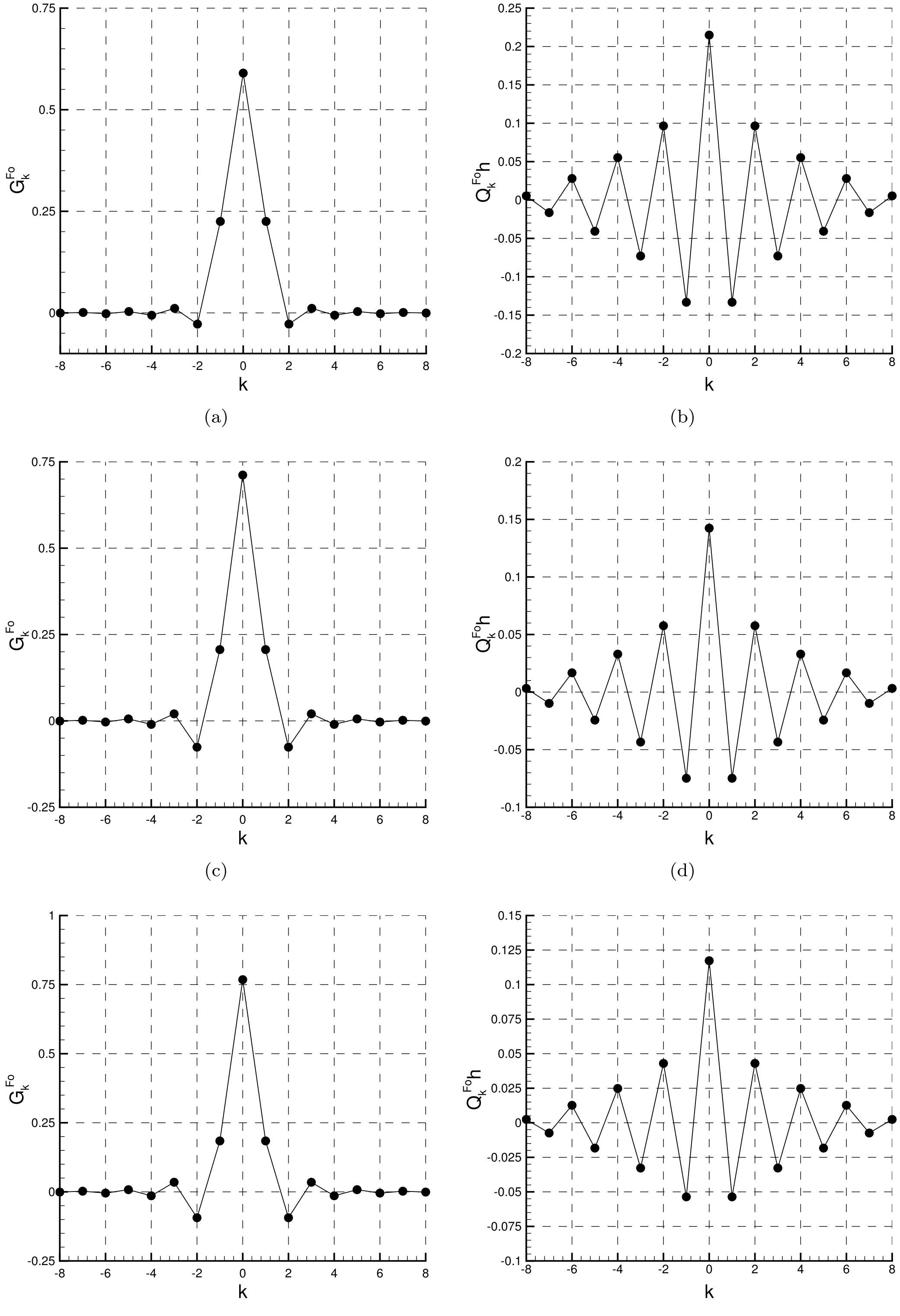
$$\delta_x^2\left(u_i\right) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \tag{53}$$

This approximation can be decomposed into two first-order derivatives, i.e.,

$$\delta_x^2\left(u_i\right) = \frac{1}{h^2} D_x(d_x\left(u_i\right)) \tag{54}$$

where

$$d_x\left(u_i\right) = u_{i+1} - u_i$$



(a) (b)

(c) (d)

(e) (f)

Fig. 6. The kernels for 2nd (a, b), 4th (c, d) and 6th (e, f) order central differencing methods with implied filter evaluated using Fourier integration. On the left, the implied numerical filters are shown (a, c, e) and on the right the corresponding Wiener inverse filter (b, d, f).

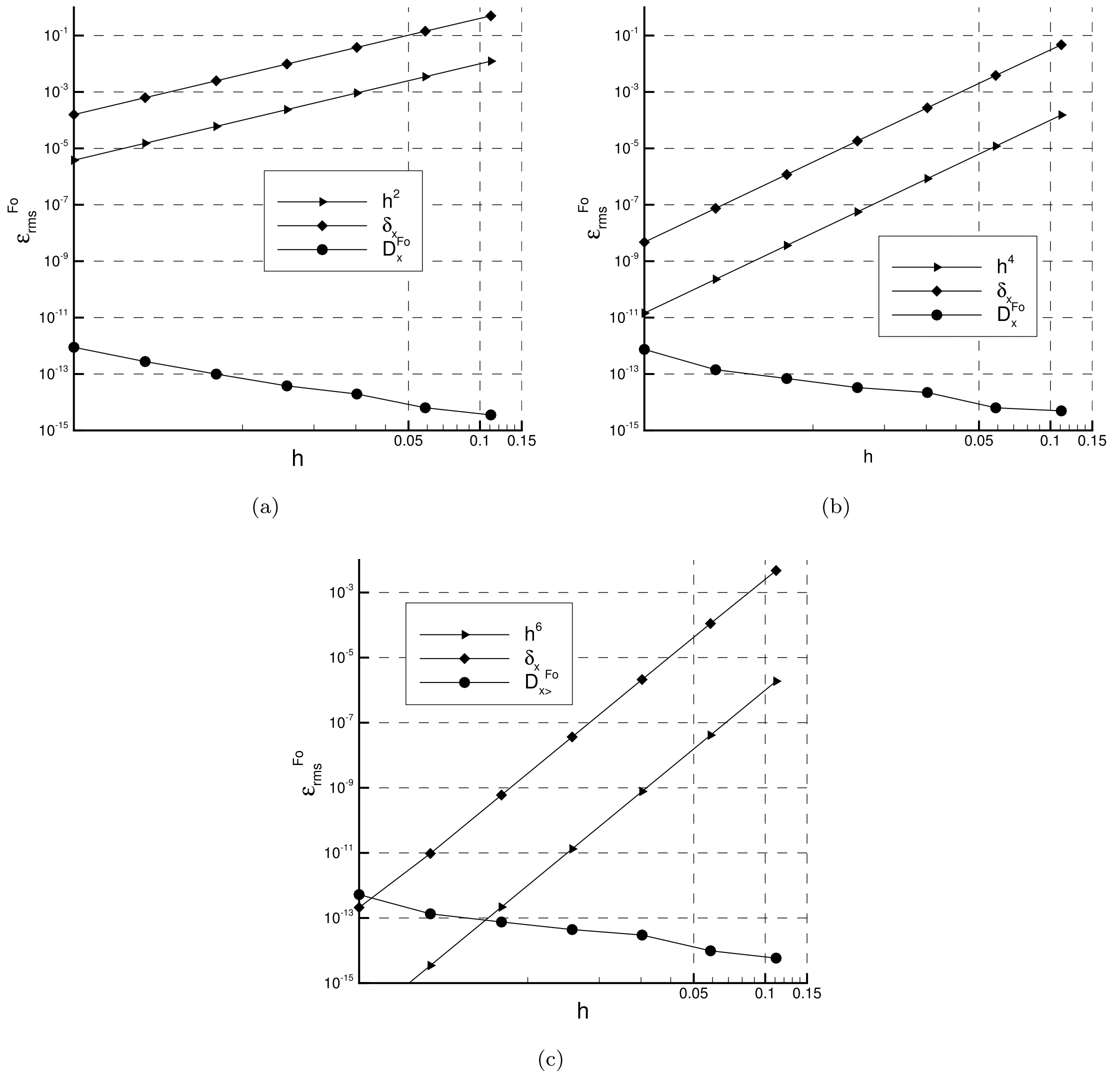


Fig. 7. Error of differentiation of the first derivative using (a) 2nd, (b) 4th and (c) 6th order basic discretisation and Fourier quadrature to approximate the implied filter. The central discretisation δ_x on 3 (a), 5 (b) and (7) grid points and the sharpened discretisation D_x^{Fo} are compared.

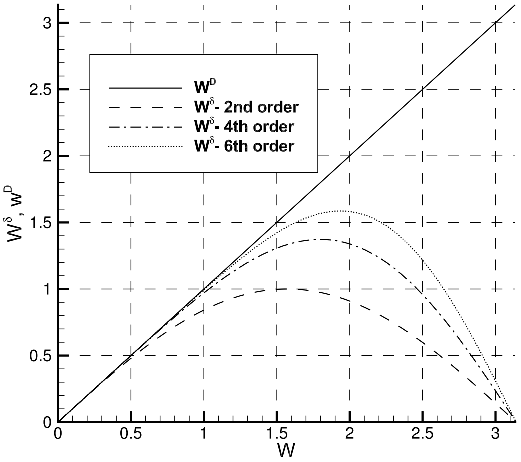


Fig. 8. Modified wavenumber for 2nd, 4th, and 6th order finite differencing and the corresponding sharpened scheme based on an implied filter defined in terms of Fourier quadrature.

$$D_x\left(u_i\right) = d_x\left(u_i\right) - d_x\left(u_{i-1}\right) \tag{55}$$

Using the method described in the previous subsection one can determine the implied filters for both first derivatives in Eq. (54), i.e., L_D, L_d and their inverses L_D^{-1}, L_d^{-1} that suggest a potentially high-order approximation of the second derivative

$$D_x^2\left(u_i\right) \equiv \frac{1}{h^2}\left(D_x \circ L_D^{-1}\right)\left(d_x \circ L_d^{-1}\left(u_i\right)\right) \approx \partial_x^2 u_i \tag{56}$$

Numerous extensions of these discrete operators can be introduced for the 4th and 6th order central approximations of the second derivative. Here, we restrict to ‘sharpened’ schemes obtained with Fourier integration of the second-order scheme (53). The Fourier series approach was applied to derive the discrete kernels according to formula (52). The discrete filter kernels for the second derivative and their inverse counterparts are shown in Fig. 9. An intricate pattern with non-zero values across the domain is obtained that yield nearly exact numerical second derivatives.

The error of second differentiation of the smooth function (34) is shown in Fig. 10. It is seen that similarly to the first derivative spectral accuracy is achieved also here. The slight increase of the error with increasing resolution stems from round-off error. Spectral resolution by the sharpened scheme for the second derivative with Fourier series interpolation is illustrated in Fig. 11. This matches the earlier result for the first derivative and establishes that spectral resolution can be realised in combination with Fourier series interpolation.

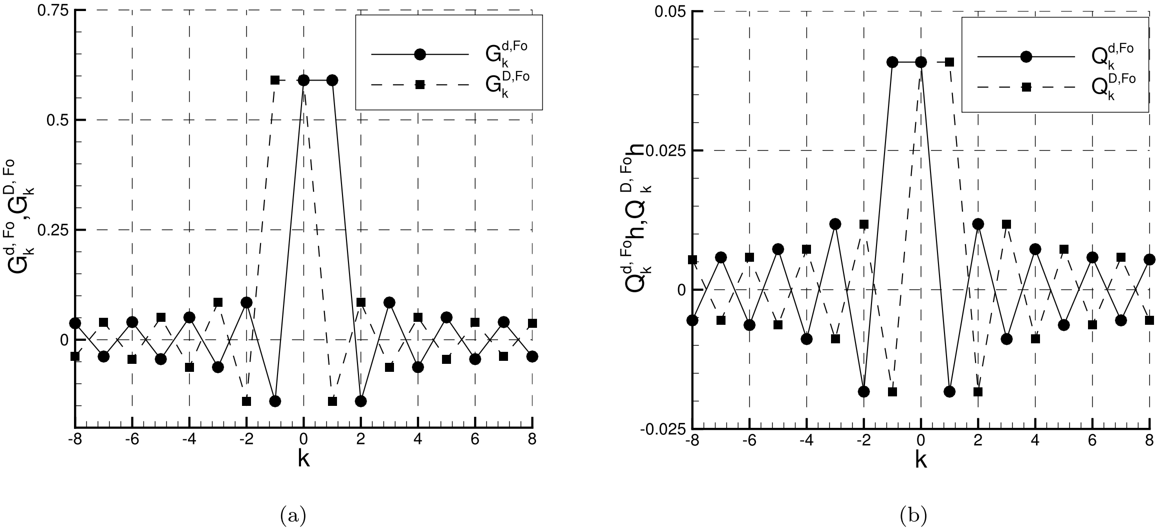


Fig. 9. The first-order kernels in the decomposition of the second order central differencing of the second derivative evaluated using Fourier integration: (a) induced numerical filter, (b) its Wiener inverse filter.

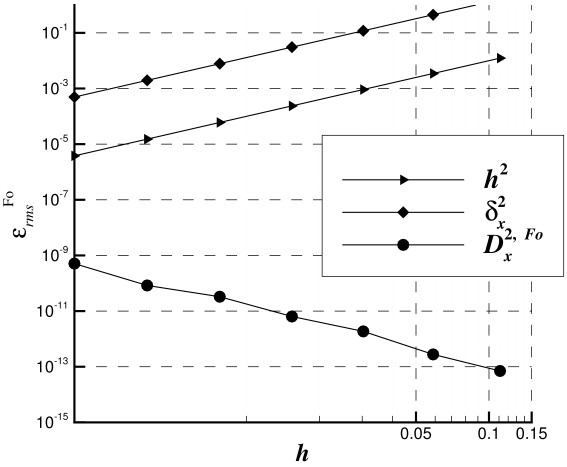


Fig. 10. Differentiation error for the second derivative, 2nd order central scheme, Fourier series used for interpolation.

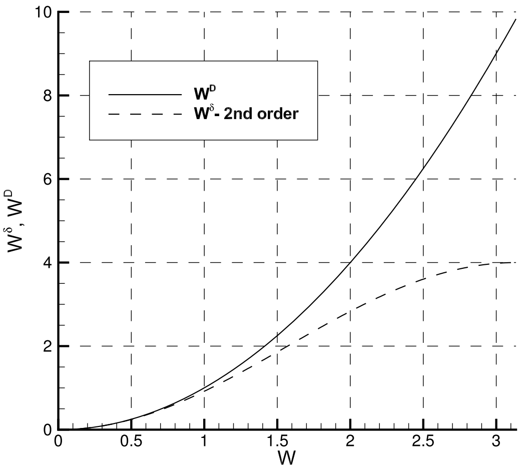


Fig. 11. Modified wave number for the sharpened of the 2nd order central scheme, Fourier series used for interpolation compared with the original scheme.

Summing up, all discrete convolution operations discussed above can be expressed in terms of matrix products, providing an alternative to the local formulation presented until now. Correspondingly, the sharpened differencing operators for the first - and second derivatives can be expressed as follows:

$$\mathbf{D} = \mathbf{\Delta Q} \tag{57}$$

$$\mathbf{D}^2 = \mathbf{\Delta}^2\mathbf{Q}^\mathbf{D}\mathbf{Q}^\mathbf{d} \tag{58}$$

Here, the following notation was introduced: **D** - sharpened scheme, **Δ** - finite differencing method and **Q** the inversion matrix for the first derivative. Likewise, for the second derivative $\mathbf{D^2}$ denotes the sharpened scheme, $\mathbf{\Delta^2}$ the finite differencing method and $\mathbf{Q^D}, \mathbf{Q^d}$ - inversion matrices of the operators in (56) for the second derivative. This matrix formulation is actually convenient for an efficient implementation of the ADD approach to sharpening a basic discretization method. Please note that in the formulation above $\mathbf{D^2}$ refers to the second derivative and not the application of **D** twice - this is in line with the chosen local notation above.

4. Numerical benefits of sharpened discretisation

In the previous sections we formulated the framework for the sharpening of low-order finite differencing to arrive at methods that can even yield spectral accuracy. In this section, based on classical second order central discretisation schemes, we establish the benefits and costs associated with the sharpening. Throughout, we will use Fourier integration and Wiener inversion to define the sharpened schemes. The new method will be applied to a number of well-known benchmark cases in Computational Fluid Dynamics (CFD), i.e., 1D convection-diffusion and Burgers equation, as well as the 2D Taylor-Green and double jet flows. These problems are often used to verify both the stability and accuracy of numerical methods [41,21,27,5,39,36]. In all test cases discussed below, the time integration is performed using the sixth-order Runge-Kutta method proposed by Outlaw et al. [31] involving eight intermediate steps.

We set out to illustrate the improvements that can be achieved by the ADD approach. However, such improvements also introduce additional steps in the algorithm. Hence, the issue of computational overhead is of relevance as well. Regarding the computational cost of a particular discretisation scheme the main factor is related to the evaluation of derivatives. In the numerical code used in this study the derivatives are computed by matrix-vector multiplication, i.e., $\mathbf{u}' = \mathbf{Du}$

and $\mathbf{u}'' = \mathbf{D}^2\mathbf{u}$, where $\mathbf{u}$, $\mathbf{u}'$ and $\mathbf{u}''$ are vectors composed of the values of u on the computational mesh, and the corresponding 1st and 2nd derivatives. Moreover, $\mathbf{D}$ and $\mathbf{D}^2$ are the matrix discretisation operators defined in (57)-(58). Assuming that for all discretisation methods adopted here, the matrices $\mathbf{D}$ and $\mathbf{D}^2$ are calculated in a preprocessing step, the computational cost would be the same for all finite differencing discretisation methods.

However, sharpened finite differencing methods rely entirely on fast matrix-vector multiplications, which come at a definite costs. Alternatives can be considered that in the end may be more effective. It is an open issue as to what method is finally the best. In fact, in the case of the 2nd order finite-difference method for which $\mathbf{D}$ and $\mathbf{D}^2$ are narrow band matrices, the multiplication of matrices can be implemented such that the sparse structure of the matrix is exploited. Other methods can be considered which may in the end be more effective. We mention two of these:

1. When the 6th order compact difference method is used, the direct calculation of $\mathbf{u}' = \mathbf{D}\mathbf{u}$ and $\mathbf{u}'' = \mathbf{D}^2\mathbf{u}$ can be replaced by solutions of equivalent three-diagonal systems of equations [26,43]. For sufficiently dense meshes, this approach is more effective than direct multiplication of the involved matrices.
2. In the case of the pseudo-spectral method, the forward and backward FFT procedure with intermediate calculations of the derivatives in spectral space is also more effective than the finite differencing method relying on direct multiplication of matrices.

Hence, from the point of view of the computational cost, the proposed sharpened scheme is the most expensive because of the reliance on matrix-vector multiplications. This argument applies to implementations on serial computers. These days, the usage of parallel computers is the standard, which may tip the balance again. For example, the solution of three-diagonal systems of equations or performing FFT requires extensive communication between processors. It may turn out that the use of specialized procedures for matrix-vector multiplication in the end will make the difference, taking into account both the ease of the implementation of the discretisation operators and its efficiency. This computational-costs issue can be settled only by focusing on actual implementations and application to generic flow problems. This requires subtle investigation which is a subject of future research.

4.1. Linear convection-diffusion equation

As the first test case, we consider the 1D linear convection-diffusion equation given as

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{59}$$

where x and t are the spatial and temporal coordinates respectively, and c and ν are a convection velocity and diffusion coefficient respectively. Simplicity of Eq. (59) and the availability of an analytic solution make this model equation very suitable as a benchmark problem, e.g., [39, 36,44]. For an unbounded domain the analytic solution takes the form of an evolving Gaussian pulse [39] defined by

$$u(x,t) = \sqrt{\frac{\sigma}{\sigma + 4\nu t}}\exp\left(-\frac{(x - x_0 - c\,t)^2}{\sigma + 4\nu t}\right) \tag{60}$$

where x_0 is the position of the pulse at $t = 0$ and σ is a parameter introduced to avoid a singularity at $t = 0$. It also sets the initial width of the Gaussian pulse. Here, we take $\sigma = 0.1$. The computational domain spans $[-2\pi, 2\pi]$ and is discretized by computational meshes consisting of $N = 33, 65, 129, 257, 513$ and 1025 nodes. The width of the initial pulse is much smaller than the width of the domain to introduce a sharply peaked initial condition, clearly within the bounds of the finite domain that is adopted in the simulations. We work with periodic conditions in the simulations - as long as the Gaussian pulse is well localized within the domain, this should be an adequate representation of the analytic solution in the unbounded domain.

We analyse the solution initially centred at $x_0 = 0$ with $c = 10$, $\nu = 0.01$ and $\nu = 0.5$, and integrate it up to the time $t = 0.2$ with a fixed time-step of $\Delta t = 1.0 \times 10^{-4}$. On the densest mesh, the Courant-Friedrichs-Lewy and von Neumann numbers (CFL $= c\Delta t/h$ and VNN $= \nu\Delta t/h^2$) are equal to CFL $= 0.08$ and $\text{VNN}_{\nu=0.01} = 0.0066$, $\text{VNN}_{\nu=0.5} = 0.33$. The stability condition for the first-order explicit Euler time integration scheme combined with second-order central finite difference spatial discretisation (denoted by δ ($\mathcal{O}(h^2)$)) is VNN < 0.5 and $\text{CFL}^2 \leq 2\text{VNN}$ [39]. In the simulations we will also consider a higher-order Runge-Kutta time integration. Since the stability region of Runge-Kutta methods is larger than that of the Euler scheme, the time-step $\Delta t = 1.0 \times 10^{-4}$ allows stable computations on all the meshes and with all time integration methods adopted here. As will be shown, this holds also for reference simulations performed applying different spatial discretisations, e.g., the sixth-order compact difference scheme (CD ($\mathcal{O}(h^6)$)) [26], the pseudo-spectral method [6], and the proposed sharpened approach (D^{Fo}). We will investigate effects of changes in spatial discretisation and use the sixth-order Runge-Kutta method to ensure that the time integration error is much smaller than the spatial discretisation error. Test computations performed at $\Delta t = 1.0 \times 10^{-5}$ did not show any observable impact of the time-step on the solutions.

Fig. 12 displays the analytic and numerical solutions to the convection-diffusion equation for $\nu = 0.01$ at $t = 0.2$ on two meshes, $N = 65$ and $N = 257$ nodes. At $t = 0.2$ the evolved Gaussian pulse has become centred at $x = 2.0$. The inset subfigures show enlarged regions of the centre of the pulse and its left branch. The grey lines in Fig. 12a present some intermediate solutions obtained applying δ ($\mathcal{O}(h^2)$) discretisation. On the coarse grid, the amplitude of the pulse significantly decreases and the solution exhibits significant and nonphysical oscillations when using the $\delta(\mathcal{O}(h^2))$ scheme. The solution is numerically stable but does not comply with the maximum principle in this setting. On the denser mesh (Fig. 12b) or when a more accurate discretisation is applied (CD ($\mathcal{O}(h^6)$)) the oscillations are considerably less pronounced. When using the sharpened approach (D^{Fo}) the oscillations do not occur. The maximum of the pulse on the mesh with $N = 65$ is most accurately calculated using the D^{Fo} method, however, its value is lower than the analytic result. On the finer mesh with $N = 257$ grid points the solutions CD ($\mathcal{O}(h^6)$), D^{Fo} and the analytic prediction are practically indistinguishable. While still slightly less accurate, the maximum value and the location where the maximum is attained are computed with good accuracy also using δ ($\mathcal{O}(h^2)$) at sufficiently high resolution.

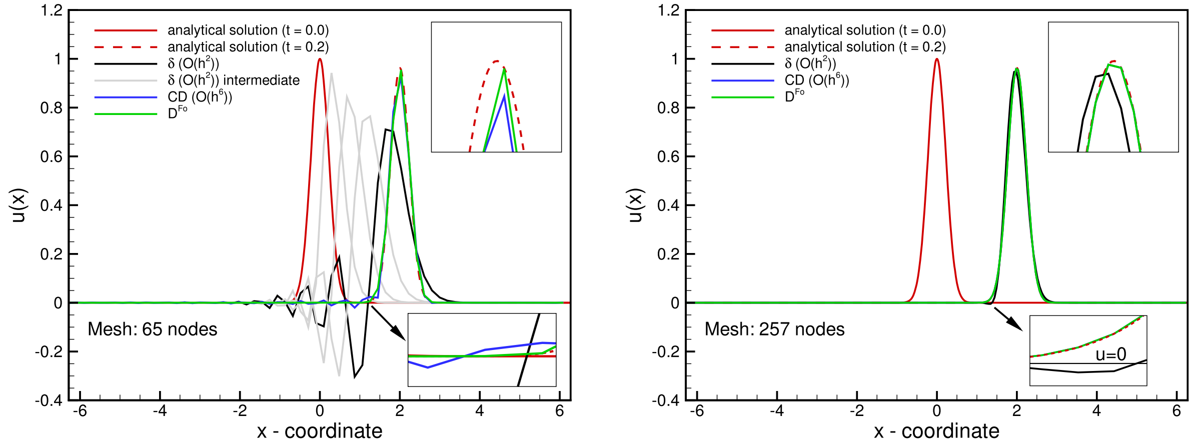
The solution error as a function of the spatial resolution is analysed in more detail next using the L_2 error norm defined as:

$$\text{Error}(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(u(x_i,t) - u^a(x_i,t))^2} \tag{61}$$

where $u^a(x_i,t)$ is the analytic solution at x_i and time t and N is the number of nodes in the grid. The achieved order of accuracy p of the applied discretisation methods (Error$\sim \mathcal{O}(h^p)$) is computed as

$$p = \frac{\log(\text{Error}_1/\text{Error}_2)}{\log(h_1/h_2)} \tag{62}$$

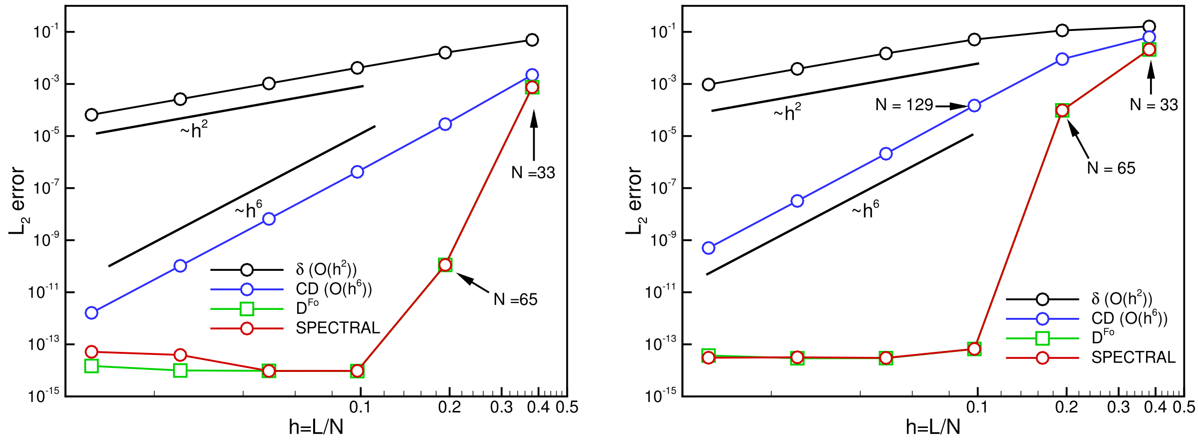
where $\text{Error}_{1,2}$ are the errors on two different meshes, with corresponding nodal spacing of h_1 and h_2. Fig. 13 shows the L_2 error norm as a function of the mesh spacing for $\nu = 0.5$ and $\nu = 0.01$, evaluating the solution at time $t = 0.2$. The theoretical error reduction with increased spatial resolution for the δ ($\mathcal{O}(h^2)$) and CD ($\mathcal{O}(h^6)$) is represented by the bold black lines. The computed values of the L_2 error and the corresponding orders of accuracy are given in Tables 1 and 2. It can be seen that when the number of nodes increases sufficiently the estimated accuracy orders asymptotically tend to the expected ones, establishing a well known text-book result. For the sharpened discretisation D^{Fo},



(a) $N = 65$

(b) $N = 257$

Fig. 12. Solution to the convection-diffusion equation for $\nu = 0.01$ at $t = 0.2$ on two meshes, $N = 65$ and $N = 257$ nodes. The analytic solutions are compared with the 2nd-order central discretisation, the 6th-order compact difference scheme, and the sharpened method. In the insets zoomed impressions of the different solutions are shown, highlighting nonphysical oscillations on coarse meshes with some of the methods (Figure a) and general convergence at sufficiently high spatial resolution (Figure b).



(a) $\nu = 0.5$

(b) $\nu = 0.01$

Fig. 13. L_2 errors (Eq. (61)) calculated for the convection-diffusion equation ($\nu = 0.5$ and $\nu = 0.01$). The arrows indicate the number of nodes corresponding to a given L/N. Second - and sixth order accuracy orders are clearly illustrated for δ ($\mathcal{O}(h^2)$) and CD ($\mathcal{O}(h^6)$). Spectral convergence is shown for both diffusion coefficients for the D^{Fo} method.

the errors closely follow those seen when adopting the pseudo-spectral method. Errors are significant on meshes with $N = 33$, $N = 65$ grid points. Beyond $N = 129$ mesh points the error level is at machine accuracy $O(10^{-14})$. Small discrepancies for $\nu = 0.5$ in the range $L/N < 0.07$ (mesh with $N \geq 513$) are due to small round-off error effects.

Independently of the ν value, the minimum number of nodes beyond which the solutions of the convection-diffusion equation reach machine precision is $N = 129$. To find the reason for such a behaviour we consider the Fourier transform of the analytic solution calculated on the mesh with $N = 1025$ nodes. We note that the number of non-zero Fourier modes is independent of ν, both for the initial solutions and at later times. Hence, we focus on the case with $\nu = 0.01$. Beyond wave number index $k = 64$ the amplitudes of the Fourier modes are of the order 10^{-12} and lower. Fig. 14 shows the amplitude spectra (FFT($u_a(x)$)) computed at $t = 0.2$ on the meshes with $N = 65$ and $N = 129$ nodes. The right axes show the error FFT(u_a) − FFT(u). One can see that it is particularly large for the method δ ($\mathcal{O}(h^2)$). On the mesh with $N = 65$ nodes, it contaminates the solution over the entire wave number range. For the CD ($\mathcal{O}(h^6)$) scheme the error is much smaller and limited mainly to wavenumbers beyond $k = 16$. Similarly, the sharpened scheme and the pseudo-spectral method display nearly identical with overlapping lines; at $N = 65$ errors can still be discriminated beyond $k = 24$, while errors are effectively 0 for $N = 129$. The error FFT(u_a) − FFT(u) assumes negative values for δ ($\mathcal{O}(h^2)$) and CD ($\mathcal{O}(h^6)$), implying an amplification of the solution compared to the analytic one. When the sharpened and pseudo-spectral methods are used at $N = 65$ a pronounced error is seen only beyond $k = 24$. This error is positive, implying that the amplitudes of the modes $k \geq 24$ are damped. In fact, even if in physical space at this coarse resolution, u is not very accurate (see Table 2) it remains smooth using D^{Fo}. On the mesh with $N = 129$ nodes, the error of the solutions obtained with $\delta(\mathcal{O}(h^2))$ and CD ($\mathcal{O}(h^6)$) is significantly reduced but still exhibits the tendency to (slightly) amplify the high-frequency components of the solution. When using the sharpened and pseudo-spectral methods, the errors are seen to drop to the round-off level. The sharpened scheme also exhibits spectral accuracy in this FFT measure. In the next section, we verify whether this holds also for a non-linear convection-diffusion equation.

Table 1
L_2 error and accuracy orders p (in brackets) of the solution of the convection-diffusion equation at $\nu = 0.5$. Calculations of accuracy orders are ill-conditioned when the error reaches machine precision.

Mesh	SPECTRAL	D^{Fo}	δ	CD
33	7.48×10^{-4} (—)	7.48×10^{-4} (—)	4.92×10^{-2} (—)	2.22×10^{-3} (—)
65	1.12×10^{-10} (23.1)	1.12×10^{-10} (23.1)	1.59×10^{-2} (1.66)	2.85×10^{-5} (6.42)
129	0.96×10^{-14} (13.6)	0.93×10^{-14} (13.7)	4.13×10^{-3} (1.96)	4.21×10^{-7} (6.15)
257	0.94×10^{-14} (—)	0.96×10^{-14} (—)	1.04×10^{-3} (2.00)	6.57×10^{-9} (6.04)
513	3.91×10^{-14} (—)	0.99×10^{-14} (—)	2.61×10^{-4} (2.00)	1.03×10^{-10} (6.01)
1025	5.17×10^{-14} (—)	1.47×10^{-14} (—)	6.54×10^{-5} (2.00)	1.61×10^{-12} (6.00)

Table 2
L_2 error and accuracy orders p (in brackets) of the solution of the convection-diffusion equation at $\nu = 0.01$. Calculations of accuracy orders are ill-conditioned when the error reaches machine precision.

Mesh	SPECTRAL	D^{Fo}	δ	CD
33	2.10×10^{-2} (—)	2.10×10^{-2} (—)	1.61×10^{-1} (—)	6.31×10^{-2} (—)
65	9.68×10^{-5} (7.94)	9.68×10^{-4} (7.94)	1.13×10^{-1} (0.52)	8.99×10^{-3} (2.87)
129	6.65×10^{-14} (30.7)	6.64×10^{-14} (30.7)	5.08×10^{-2} (1.17)	1.48×10^{-4} (5.98)
257	3.01×10^{-14} (—)	2.95×10^{-14} (—)	1.48×10^{-2} (1.78)	2.09×10^{-6} (6.17)
513	3.17×10^{-14} (—)	2.91×10^{-14} (—)	3.79×10^{-3} (1.97)	3.21×10^{-8} (6.04)
1025	3.06×10^{-14} (—)	3.72×10^{-14} (—)	9.52×10^{-4} (1.99)	5.01×10^{-10} (6.01)

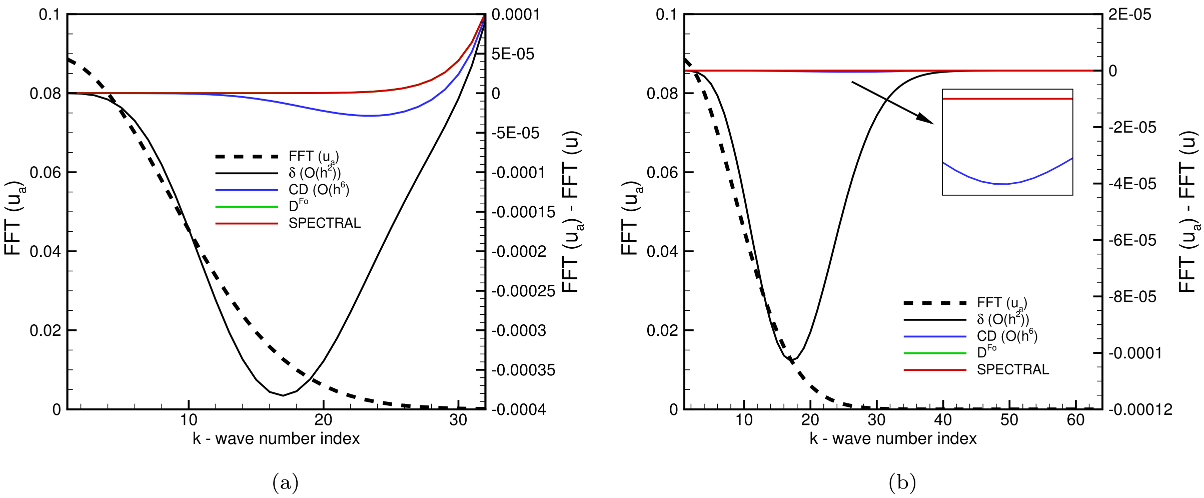


Fig. 14. Amplitude spectra of the solutions to the convection-diffusion equation ($u_a(x)$ and $u(x)$) for $\nu = 0.01$ computed at $t = 0.2$ on the meshes with $N = 65$ (a) and $N = 129$ (b) nodes.

4.2. Burgers equation

The Burgers equation is the non-linear form of the convection and diffusion equation analyzed in the previous section. This model system has been used frequently to qualitatively understand flow dynamics and the role numerical methods play in simulations. Mainly because the convection and diffusion terms appear in a divergence form in much the same way as in the Navier-Stokes equation. The Burgers is given as [3]

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{63}$$

The scalar variable u is often interpreted as a velocity component. In the periodic domain $[0, 2\pi[$ an analytic solution to the Burgers equation is obtained through the Cole-Hopf transformation [19,8]. This solution reads

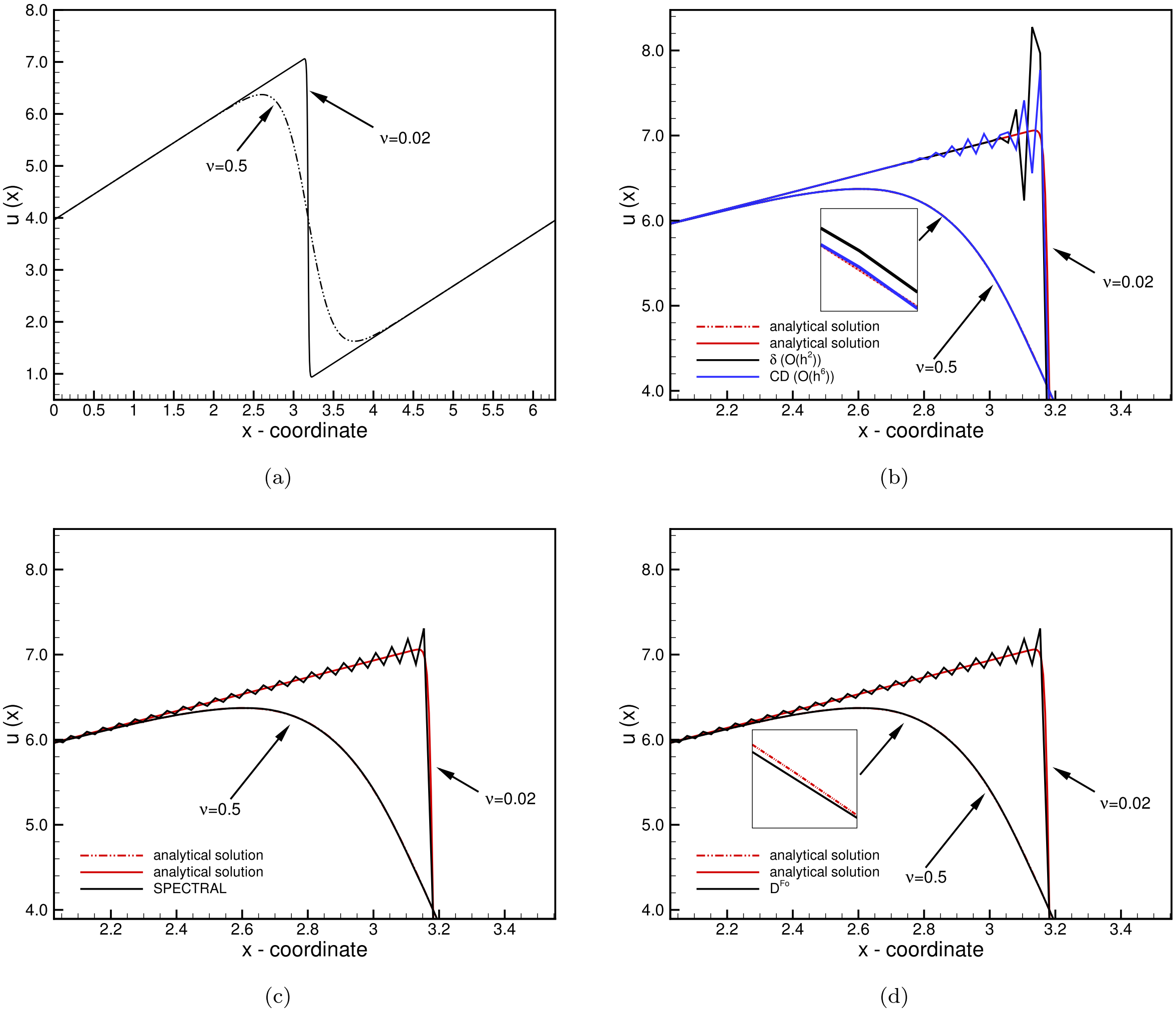
$$u(x,t) = -\frac{2\nu}{\phi(x,t)}\frac{\partial}{\partial x}\phi(x,t) + c \tag{64}$$

where c is an arbitrary constant, here taken to be $c = 4$, and the function $\phi(x,t)$ is given as

$$\phi(x,t) = \sum_{n=-\infty}^{\infty} \exp\left(-\frac{(x - ct - 2n\pi)^2}{4\nu(t+1)}\right). \tag{65}$$

We analyse the solutions with $\nu = 0.02$ and $\nu = 0.5$ integrated up to the time $t = 0.01$ with a fixed time-step of $\Delta t = 1.0 \times 10^{-5}$. The integration time is long enough to reveal spatial discretisation errors without the influence of the time integration error. To verify the independence of the solutions on the time-step, test computations have been performed at $\Delta t = 1.0 \times 10^{-6}$. As in the previous section, the meshes consisted of $N = 33, 65, 129, 257, 513$ and 1025 nodes.

The high value of ν represents strongly dissipative dynamics, while the low value of ν corresponds to low dissipation and the well-known



(a)

(b)

(c)

(d)

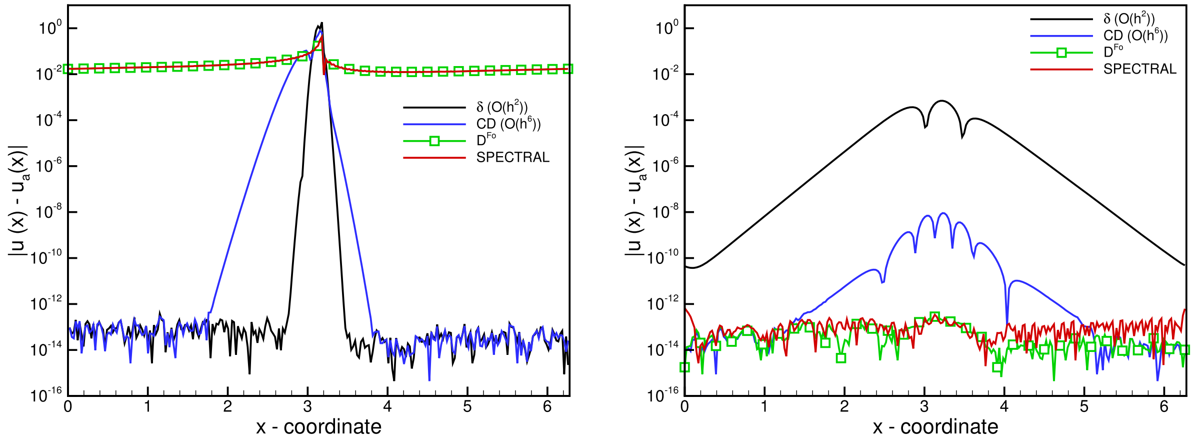
Fig. 15. Solution to the Burgers equation for $\nu = 0.02, 0.5$ at $t = 0.01$. The analytic solutions (a) are compared with the 2nd order central discretisation and the 6th-order compact difference scheme (b), the sharpened method (c) and pseudo-spectral discretisation (d).

appearance of a region with high gradient in the solution. The analytic solutions in the entire computational domain are presented in Fig. 15a. Comparisons with numerical solutions in the vicinity of the gradient are shown in the remaining subfigures, they were obtained on a grid with $N = 129$ nodes. The solution obtained using the sharpened method (D^{Fo}) is included in Fig. 15(c) for which, for clarity of the presentation, the pseudo-spectral reference solution [6] is presented in a separate subfigure (Fig. 15(d)). In the case with $\nu = 0.02$ the basic method ($\delta O(h^2)$) shows large oscillations near the sharp gradient. Similar oscillations occur when the CD-$O(h^6)$ scheme is used. In the case of the sharpened and the pseudo-spectral method the amplitudes of the oscillations are much reduced but the oscillations spread over all mesh points (Fig. 15(c,d)). This is very well seen in Fig. 16(a) presenting differences between the analytic and numerical solutions in particular points. The situation changes for the case with $\nu = 0.5$ for which the solution is smoother and can be captured accurately on the selected mesh with $N = 129$ nodes with any of the adopted numerical methods. In this case, the sharpened method shows highly competitive predictions compared to the solutions obtained using $\delta(\mathcal{O}(h^2))$ and CD ($\mathcal{O}(h^6)$) schemes. The analysis of the errors presented in Fig. 16(b) shows that its accuracy is comparable with the accuracy obtained using the spectral method. All methods included here show reduced oscillations in case the spatial resolution is further increased. This establishes correct convergence at high resolution, despite the significant oscillations at more practical numbers of grid points.

The solution errors are shown in Fig. 17. For $\nu = 0.5$ (Fig. 17(a)) convergence is well-established for $\delta(\mathcal{O}(h^2))$ and CD ($\mathcal{O}(h^6)$) at the expected theoretical rates of 2 and 6 respectively. For the scheme D^{Fo} the error is reduced in the same way as for the pseudo-spectral method. In both cases, machine precision is reached already on the mesh with $N = 129$ nodes. Further increase of the number of mesh points does not improve the solution accuracy as the round-off errors significantly affect the remaining error in the solution. However, unlike in the case of the linear convection-diffusion equation, this time the sharpened scheme D^{Fo} appears much more resistant to this type of simulation error compared to the pseudo-spectral method. Most likely because of different round-off error dynamics caused by the non-linear convection terms.

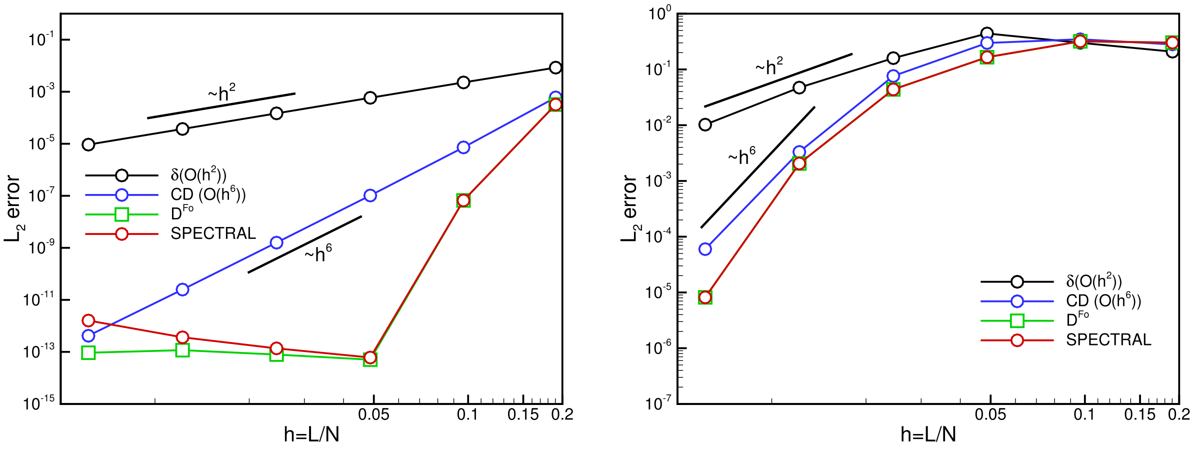
The solutions obtained for $\nu = 0.02$ are significantly less accurate on the range of selected spatial resolutions. The schemes $\delta(\mathcal{O}(h^2))$ and CD ($\mathcal{O}(h^6)$) manifest their expected convergence rates only on the densest meshes included in this study. The very pronounced gradient in the solution near the centre of the domain also makes itself known in terms of the range of resolutions for which the asymptotic convergence sets in. Regarding the method D^{Fo} and the pseudo-spectral method the computed errors are virtually identical to each other and much lower compared to the error reported for CD ($\mathcal{O}(h^6)$), although the difference is not as pronounced as for the case with $\nu = 0.5$. Also for these methods, the asymptotic convergence sets in only at more refined meshes, compared to the case $\nu = 0.5$.

To further quantify the convergence characteristics, in Table 3 and Table 4 we show the errors and computed orders of accuracy. The



(a) (b)

Fig. 16. Local solution errors to the Burgers equation for $\nu = 0.02$ (a) and $\nu = 0.5$ (b) at $t = 0.01$.



(a) $\nu = 0.5$ (b) $\nu = 0.02$

Fig. 17. L_2 errors (Eq. (61)) calculated for the Burgers equation ($\nu = 0.5$ and $\nu = 0.02$).

Table 3
L_2 errors and p-orders (in brackets) of the Burgers equation ($\nu = 0.5$).

Mesh	SPECTRAL	D^{Fo}	δ	CD
33	3.21×10^{-4} (—)	3.21×10^{-4} (—)	8.43×10^{-3} (—)	6.03×10^{-4} (—)
65	6.62×10^{-8} (12.5)	6.62×10^{-8} (12.5)	2.27×10^{-3} (1.93)	7.24×10^{-6} (6.52)
129	6.07×10^{-14} (20.2)	5.02×10^{-14} (20.5)	5.83×10^{-4} (1.98)	1.02×10^{-7} (6.21)
257	1.36×10^{-13} (—)	7.89×10^{-14} (—)	1.47×10^{-4} (1.99)	1.58×10^{-9} (6.04)
513	3.63×10^{-13} (—)	1.16×10^{-13} (—)	3.70×10^{-5} (1.99)	2.48×10^{-11} (6.01)
1025	1.60×10^{-12} (—)	9.30×10^{-14} (—)	9.27×10^{-6} (1.99)	4.17×10^{-13} (5.90)

Table 4
L_2 errors and p-orders (in brackets) of the Burgers equation ($\nu = 0.02$).

Mesh	SPECTRAL	D^{Fo}	δ	CD
33	3.02×10^{-1} (—)	3.02×10^{-1} (—)	2.08×10^{-1} (—)	2.81×10^{-1} (—)
65	3.18×10^{-1} (-0.07)	3.18×10^{-1} (-0.07)	3.00×10^{-1} (-0.53)	3.47×10^{-1} (-0.31)
129	1.65×10^{-1} (0.95)	1.65×10^{-1} (0.95)	4.41×10^{-1} (-0.56)	2.98×10^{-1} (0.22)
257	4.37×10^{-2} (1.93)	4.37×10^{-2} (1.93)	1.59×10^{-1} (1.47)	7.62×10^{-2} (1.97)
513	2.05×10^{-3} (4.42)	2.05×10^{-3} (4.42)	4.71×10^{-2} (1.76)	3.31×10^{-3} (4.53)
1025	8.16×10^{-6} (7.98)	8.16×10^{-6} (7.98)	1.02×10^{-2} (2.20)	5.98×10^{-5} (5.79)

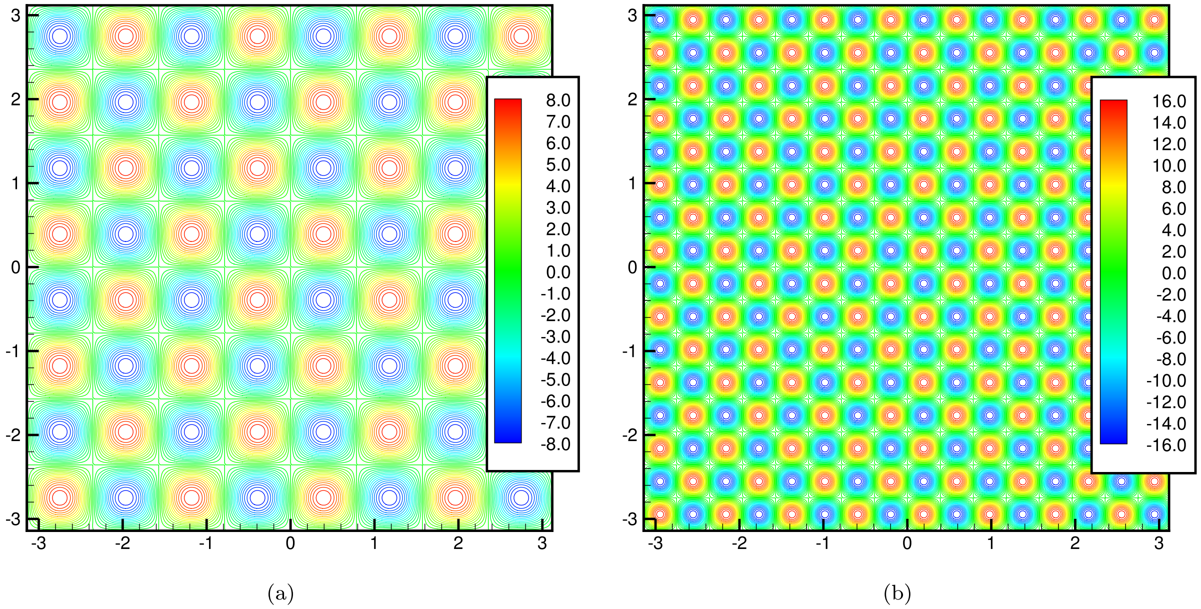


Fig. 18. Initial vorticity contours in the Taylor-Green flow with the parameter $T_v = 4$ (a) and $T_v = 8$ (b).

pseudo-spectral and sharpened discretisation correspond very closely to each other in terms of L_2-norm of the error. These results illustrate that only at sufficiently high spatial resolutions one can benefit from the formal spectral convergence rate.

4.3. Taylor-Green flow

For the investigation of the sharpened discretisation method in two spatial dimensions we consider the well-known Taylor-Green flow. We consider the 2D problem and describe the flow in the vorticity–stream function formulation (ω, ψ) that is often used in CFD to test the accuracy and stability of numerical schemes [2,38,37,54,5]. It is defined as

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \tag{66}$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \tag{67}$$

where $u = \partial_y \psi$, $v = -\partial_x \psi$ are the velocity components, and ν is the kinematic viscosity, which we set $\nu = 0.001$ to create a challenging case with low dissipation and a wide range of dynamical scales. In two dimensions the vorticity is given by $\omega = \partial_x v - \partial_y u$. The (ω, ψ) formulation inherently fulfils the requirement that the velocity field (u, v) should be divergence-free. The Poisson equation (67) is solved using the matrix diagonalization technique [6,43]. In all analyzed cases we set the time-step Δt significantly lower than the stability limit

$$\Delta t < \alpha \Delta t_s \quad ; \quad \Delta t_s = C\frac{h}{\max_{i,j}(|u_{i,j}| + |v_{i,j}|) + 2\nu/h} \tag{68}$$

with $C = 1.8/\pi$ [18] ensuring stable computations for the pseudo-spectral method, which from the point of view of the time-step limit is the most restrictive. The factor $\alpha = 0.1$ is adopted to reduce residual errors associated with the time integration method - here, we are interested in errors due to the spatial discretisation and therefore in the simulations largely eliminate errors due to time integration.

The Taylor-Green flow is one of a few problems for which an analytic solution of the Navier-Stokes equations is available. Thus, this test case enables a precise examination of the accuracy of CFD algorithms and discretisation schemes. Among others, it was used as the benchmark case in [24,27,43]. In terms of the horizontal and vertical velocity components (u, v) the analytic solution is given by

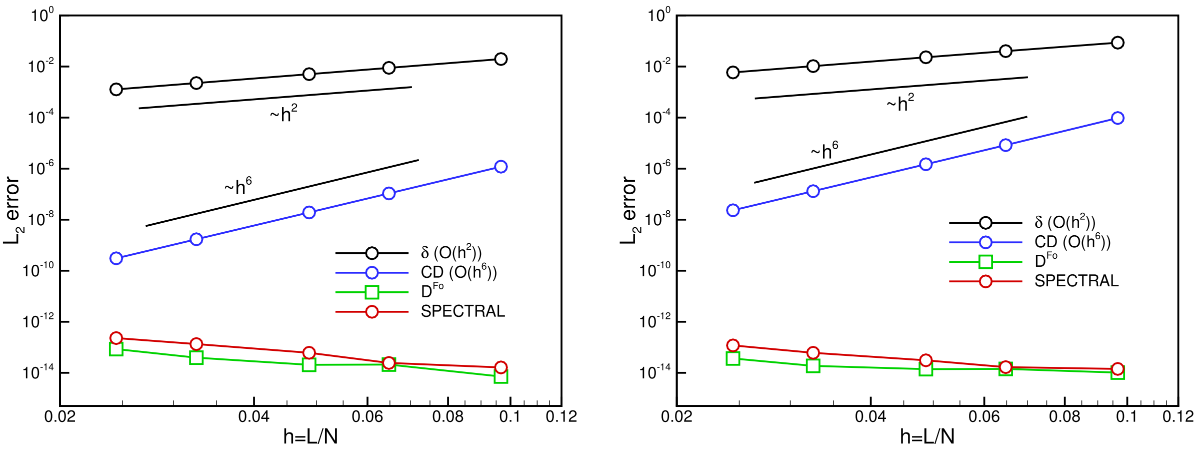
$$\begin{aligned} u(x,y,t) &= -\sin(T_v x)\cos(T_v y) \times \exp\left(-2\nu T_v^2 t\right) \\ v(x,y,t) &= \cos(T_v x)\sin(T_v y) \times \exp\left(-2\nu T_v^2 t\right) \end{aligned} \tag{69}$$

where T_v denotes the number of vortices in the periodic domain. The larger it is the smaller the vortices, and thus, their accurate numerical solution becomes more demanding. Here, we adopt $T_v = 4$ and $T_v = 8$ to define different test cases. The initial solutions in a square domain $L = [-\pi, \pi]^2$ are presented in Fig. 18. For ν equal to zero the vortices should not be decaying at all, provided that the numerical method is dissipation-free. Here, we assume $\nu = 0.001$ which leads to a very slow temporal decrease of the initial amplitude of the vortices. The computations are performed up to time $t = 0.1$, which is sufficient to reveal accuracy differences between particular discretisation methods. The computational meshes consist of $N^2 = 65^2, 97^2, 129^2, 193^2, 257^2$ nodes. The time-step equals $\Delta t = 1 \times 10^{-4}$ and we checked that lowering Δt does not change the results in any significant way. The solution error in 2D is defined analogously to (61).

For both T_v values, the coarsest mesh ensures a resolution with at least 8 grid cells per vortex, to capture all initial flow structures with accuracy. This spatial resolution remains adequate as the Taylor-Green vortices only dissipate and smaller scale flow features do not emerge. Hence, based on the analysis of the convection-diffusion equation in spectral space, one could expect that using the sharpened and pseudo-spectral methods should (nearly) coincide with the exact solution. Indeed, this is what is found in Fig. 19 showing the error of the u-velocity component as function of the number of nodes. On all meshes the error of the sharpened and pseudo-spectral methods coincide at machine accuracy level and agree to within the round-off error. As in the 1D cases, the round-off error slightly grows when h decreases. The errors in the solutions obtained with δ ($\mathcal{O}(h^2)$) and CD ($\mathcal{O}(h^2)$) decrease perfectly in line with the corresponding discretisation orders. The error values at $T_v = 8$ are slightly larger when applying δ ($\mathcal{O}(h^2)$) and CD ($\mathcal{O}(h^2)$) in the higher wavenumber range (see Figs. 8, 11) - this is due to lower spatial resolution per vortex in this case.

4.4. Double-jet flow

Finally, we turn to the double jet problem, the dynamics of which results from the interactions of small and large flow structures. The double-jet flow is an excellent test case for verifying the accuracy of the discretisation method. This flow is characterized by a wide range



(a)

(b)

Fig. 19. L_2 errors calculated for the Taylor-Green flow with the parameter $T_v = 4$ (a) and $T_v = 8$ (b).

of length - and time scales. Similarly, as for the Taylor-Green flow, we assume a doubly-periodic domain $[0, 2\pi] \times [0, 2\pi]$. The initial velocity u in the horizontal direction is defined as

$$u(x, y, 0) = \begin{cases} \tanh\left(\frac{y-\pi/2}{\delta}\right) & \text{for} \quad y \le \pi \\ \tanh\left(\frac{3\pi/2-y}{\delta}\right) & \text{for} \quad y > \pi \end{cases} \tag{70}$$

and the initial vertical velocity component v is taken as

$$v(x, y, 0) = \epsilon \sin(x) \tag{71}$$

The strength of the initial velocity disturbance is determined by ϵ. Moreover, the parameter δ is the shear layer thickness. Here, similarly as in [21] we take $\epsilon = 0.05$ and $\delta = \pi/15$. The initial velocity field forms two parallel shear layers, the upper one with negative vorticity and the lower one with positive vorticity. An illustration of the evolving vorticity field is provided in Fig. 20). For these simulations the new sharpened spatial discretisation method was adopted. At the start of the simulation, the growing initial perturbation leads to the roll-up of these shear layers and the formation of large vortices. As time passes, the shear layers between these vortices stretch and become thinner. At low viscosity values, as selected here, small flow structures occur and the representation of the solution on meshes that yield feasible simulations requires a highly accurate discretisation method. Hence, this test-case is well suited for the assessment of the new sharpened discretisation method.

The simulations were performed on two meshes with different resolution, i.e., a relatively coarse mesh with 129^2 nodes (mesh M1) and a considerably finer mesh with 513^2 nodes (mesh M2). The time step is taken as $\Delta t = 5 \times 10^{-4}$ on both meshes. The vorticity contours initially develop shear layers with a wavy shape (time $t = 1.0 - 3.0$). This wavy pattern grows with time and the vortices fully roll up in between $t = 7.0$ and $t = 9.0$. After this roll-up phase, the structures in the flow remain quite similar over a long time as the flow gradually decays under the influence.

The double-jet test case has no known analytic solution. Therefore, to measure the error induced by a particular discretisation method, we focus on the palinstrophy ($\mathcal{P}(t)$). This quantity is defined as

$$\mathcal{P}(t) = \frac{1}{2} \int_0^{2\pi} \int_0^{2\pi} (\nabla \omega(x, y, t))^2 \mathrm{d}x\mathrm{d}y \tag{72}$$

The error induced by one of the discretisation methods is quantified by comparing the predicted palinstrophy to the results obtained using the pseudo-spectral method on the same mesh, i.e.: $\mathcal{P}(t) - \mathcal{P}^{PS}(t)$. This measure accounts for both the velocity components and their gradients over the whole domain. Additionally, the maxima of the velocity components (MAX($u_{i,j}(t)$), MAX($v_{i,j}(t)$)) are compared with the values obtained using the pseudo-spectral method to further classify the accuracy.

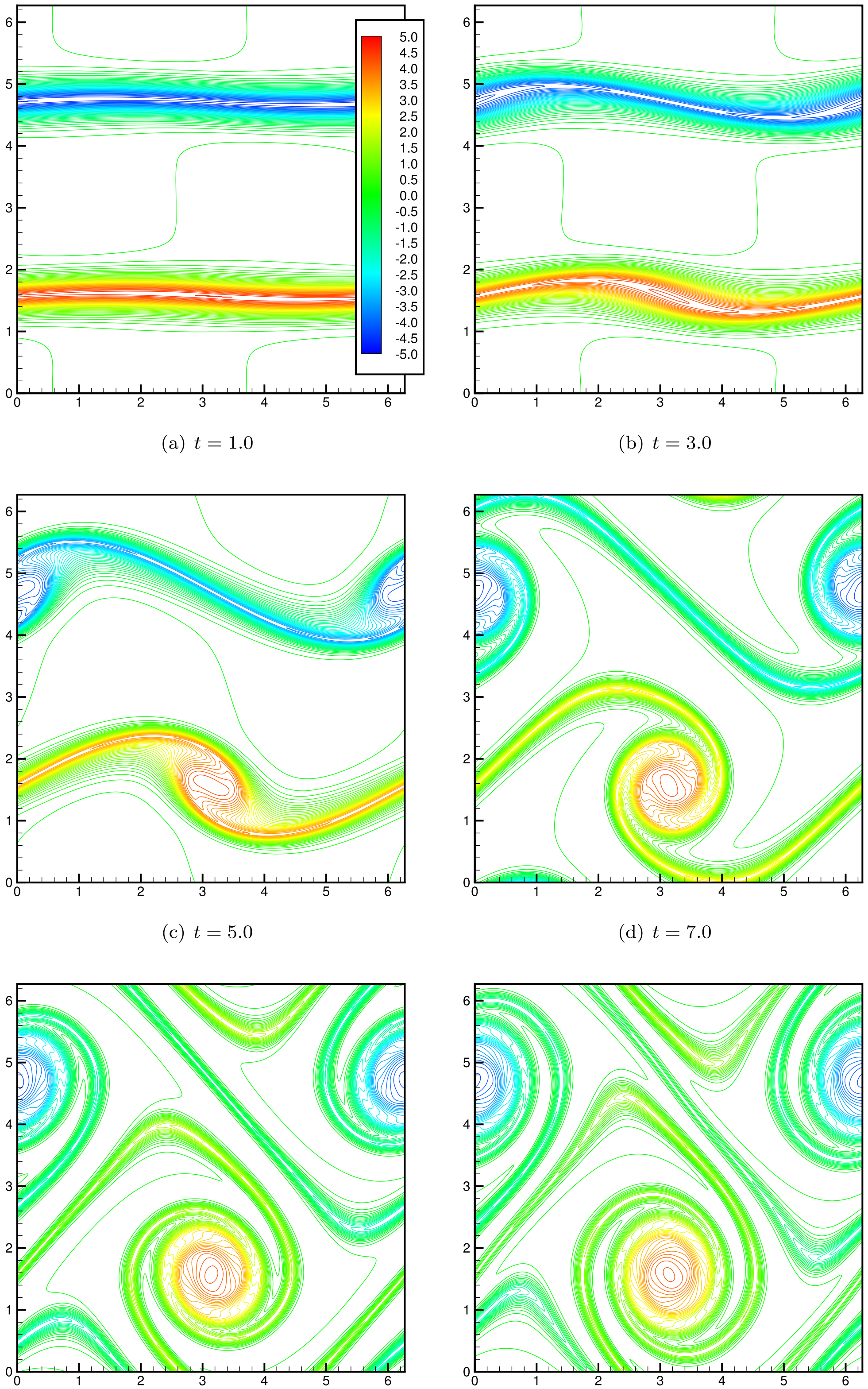
Fig. 21 shows the temporal evolution of the palinstrophy and the maxima of the velocity components obtained on the mesh with M1. During the initial stages of the simulation $t \lesssim 4$ the vorticity gradient decreases, which manifests itself by a lowering of $\mathcal{P}(t)$. During this phase, a rapid increase of the vertical and horizontal velocity components is observed, which continues until the maximum value in velocities as well as palinstrophy is reached around $t = 9.0$. Beyond this time, the vortical structures are fully rolled-up. It can be seen that the errors are considerable for the basic scheme $\delta(\mathcal{O}(h^2))$. In the case of the CD ($\mathcal{O}(h^6)$) scheme, the errors are much reduced and their maximum values occur in the roll-up phase $5 < t < 10$. Evolutions of the solution error of palinstrophy on the meshes M1 and M2 are shown in Fig. 22(a,b). Increasing the number of nodes four times (M1→M2) reduces the errors approximately 16 and 4096 times for the 2nd and 6th order scheme, respectively. Hence, the errors in the simulations with CD ($\mathcal{O}(h^6)$) scheme are very small, and in fact, the obtained solution is close to the pseudo-spectral one. Regarding the accuracy of the sharpened method (D^{Fo}) the solutions correspond very closely to the ones obtained using the pseudo-spectral method. The errors that are virtually equal to zero regardless of the mesh density and the flow phase. This confirms that the D^{Fo} scheme provides the same accuracy as the pseudo-spectral method.

5. Conclusions

A new method to construct discretisation schemes was presented. The philosophy behind this construction is to anticipate discretisation errors that occur with any standard finite differencing method and correct for these as integral part of the new method. This is referred to as 'sharpening' in which the implied filter of associated with a basic differencing method is approximately inverted.

The construction proceeds in a few steps. Knowing the explicit form of the implied filter [15], first, its discrete form is calculated, after which its inverse form can be approximated. In the present paper, van Cittert and Wiener type inverse filters were considered. In simulations of flow in periodic domains the Wiener inversion was preferred as it yields the exact inverse filter kernel corresponding to the discrete implied filter on a given mesh.

It was shown that the accuracy of the new scheme is determined by the form of the discrete implied kernel. To improve the accuracy of the 'sharpened' scheme relative to the original scheme, the quadrature of



(a) $t = 1.0$

(b) $t = 3.0$

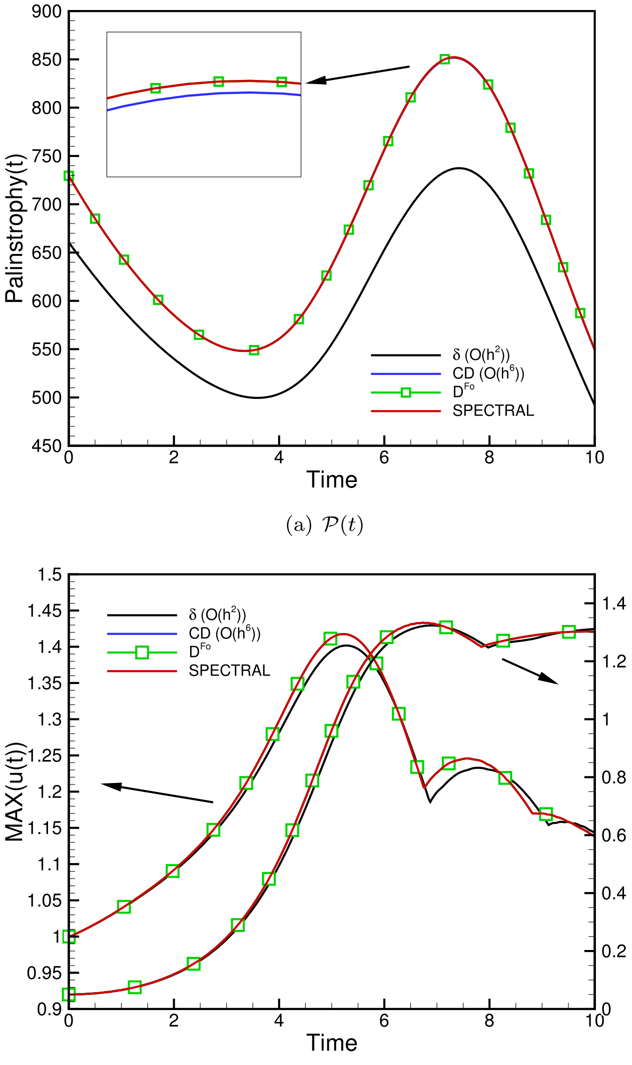
(c) $t = 5.0$

(d) $t = 7.0$

(e) $t = 9.0$

(f) $t = 11.0$

Fig. 20. Vorticity contours in the double jet flow. The colour scale ranges from -5 (blue) to 5 (red).



(a) $\mathcal{P}(t)$

(b) $\mathrm{MAX}(u_{i,j}(t))$, $\mathrm{MAX}(v_{i,j}(t))$

Fig. 21. Palinstrophy (a) and maximum velocity evolutions (b) on the mesh M1.

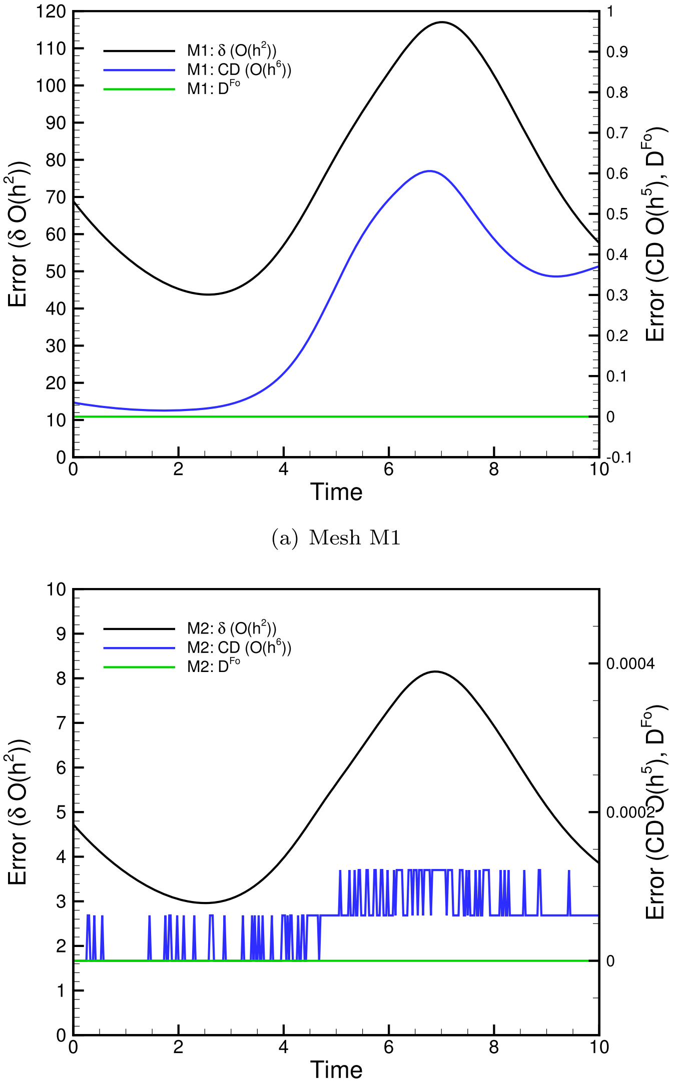
the implied filter must be of higher order than the order of accuracy of the basic differencing method. Using the Fourier series for the quadrature the new 'sharpened' scheme shows spectral accuracy comparable to the pseudo-spectral approximation.

The new scheme using the Fourier series quadrature was tested for the linear convection-diffusion equation, the Burgers equation, the Taylor-Green flow, and for double-jet flow predicted by the Navier-Stokes equations. The results were compared with standard second-order differences, compact differences of sixth-order and the Fourier pseudo-spectral method. The results confirmed that the new scheme has spectral accuracy.

The current implementation of the proposed numerical scheme applies to periodic problems. More general boundary conditions can be considered, replacing the Wiener inverse filter with a van Cittert procedure and high-order (Cavalieri-Simpson) quadrature. This will no longer show spectral accuracy, but will enable application of the sharpening method to general non-periodic problems. Current research is dedicated to this extension.

Data availability

Data will be made available on request.



(a) Mesh M1

(b) Mesh M2

Fig. 22. Evolution of the solution error of palinstrophy on the meshes M1 (a) and M2 (b).

Acknowledgements

This work was supported by the National Science Centre, Poland (Grant no. 2018/29/B/ST8/00262) and statutory funds of Czestochowa University of Technology BS/PB 1-100-3011/2022/P. The international collaboration with Prof. Bernard J. Geurts was possible thanks to the ANIMATE project (Grant no. PPI/APM/2019/1/00062/DEC/01) funded by Polish National Agency for Academic Exchange.

References

[1] W. Aniszewski, A. Boguslawski, M. Marek, A. Tyliszczak, J. Comput. Phys. 231 (2012) 7368–7397.
[2] G. Ashcroft, X. Zhang, J. Comput. Phys. 190 (2003) 459–477.
[3] J.M. Burgers, Adv. Appl. Mech. 1 (1948) 171–199.
[4] L.C. Berselli, T-Y. Kim, L.G. Rebholz, Discrete Contin. Dyn. Syst., Ser. B 21 (2015) 1027–1050.
[5] L. Caban, A. Tyliszczak, Comput. Math. Appl. 1089 (2022) 123–140.
[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, 1988.
[7] P. Campisi, K. Egiazatian (Eds.), Blind Image Deconvolution, CRC Press, 2017.
[8] J.D. Cole, Q. Appl. Math. 9 (1951) 225–236.
[9] J.A. Domaradzki, E.M. Saiki, Phys. Fluids 9 (1997) 2148–2164.
[10] P. Domingo, L. Vervisch, Combust. Flame 177 (2017) 109–122.

[11] B.J. Geurts, Phys. Fluids 9 (1997) 3585–3587.
[12] B.J. Geurts, J. Turbul. (2006) N55.
[13] B.J. Geurts, D.D. Holm, Alpha-Modeling Strategy for LES of Turbulent Mixing, Springer, 2002.
[14] B.J. Geurts, A. Rouhi, U. Piomelli, J. Fluids Struct. 91 (2019) 102615.
[15] B.J. Geurts, F. van der Bos, Phys. Fluids 17 (2005) 125103.
[16] B.J. Geurts, Elements of Direct and Large-Eddy Simulation, R.T. Edwards Inc., 2003.
[17] F.F. Grinstein, C. Fureby, C.R. DeVore, Int. J. Numer. Methods Fluids 47 (2005) 1043–1051.
[18] B. Gustafsson, H.-O. Kreiss, J. Oliger, Time Dependent Problems and Difference Methods and Sprays, John Wiley and Sons, 1995.
[19] E. Hopf, Commun. Pure Appl. Math. 3 (1950) 201–230.
[20] P. Iannelli, F.M. Denaro, G. De Stefano, Int. J. Numer. Methods Fluids 43 (2003) 431–462.
[21] R. Knikker, Int. J. Numer. Methods Fluids 59 (2009) 1063–1092.
[22] E.M.J. Komen, L.H. Camilo, A. Shams, B.J. Geurts, B. Koren, J. Comput. Phys. 345 (2017) 565–595.
[23] J.G.M. Kuerten, B.J. Geurts, A.W. Vreman, M. Germano, Phys. Fluids 11 (1999) 3778.
[24] S. Laizet, E. Lamballais, High-order compact schemes for incompressible flows: a simple and efficient method with quasi-spectral accuracy, J. Comput. Phys. 228 (2009) 5989–6015.
[25] W. Layton, Comput. Methods Appl. Mech. Eng. 200 (2011) 3183–3199.
[26] S.K. Lele, J. Comput. Phys. 103 (1992) 16–42.
[27] M.L. Minion, D.L. Brown, J. Comput. Phys. 138 (1997) 734–765.
[28] J. Meyers, B.J. Geurts, M. Baelmans, Phys. Fluids 15 (2003) 2740–2755.
[29] C. Mehl, J. Idier, B. Fiorina, Combust. Theory Model. 22 (2018) 38–70.
[30] Z.M. Nikolaou, L. Vervisch, Flow Turbul. Combust. 101 (2018) 33–53.
[31] C. Outlaw, L. Derr, D. Sarafyan, Comput. Math. Appl. 12 (1986) 815–824.
[32] U. Piomelli, A. Rouhi, B.J. Geurts, Phys. Rev. Fluids 1 (2016) 044401.
[33] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer, 2007.
[34] A. Rouhi, U. Piomelli, B.J. Geurts, J. Fluid Mech. 766 (2015) 499–527.
[35] P. Sagaut, Large Eddy Simulation for Incompressible Flows - An Introduction, Springer, 2006.
[36] P. Sagaut, V.K. Suman, P. Sundaram, M.K. Rajpoot, Y.G. Bhumkar, S. Sengupta, A. Sengupta, T.K. Sengupta, Comput. Fluids 261 (2023) 105915.
[37] T.K. Sengupta, N. Sharma, A. Sengupta, Phys. Fluids 30 (2018) 054105.
[38] T.K. Sengupta, A. Sengupta, J. Comput. Phys. 310 (2016) 1–25.
[39] V.K. Suman, T.K. Sengupta, C.J.D. Prasad, K. Surya Mohan, Deepanshu Sanwalia, Comput. Fluids 150 (2017) 95–114.
[40] F. Schwertfirm, J. Mathiew, M. Manhart, Comput. Fluids 37 (2008) 431–462.
[41] A. Shah, L. Yuan, A. Khan, Appl. Math. Comput. 215 (2010) 3201–3213.
[42] S. Stoltz, N.A. Adams, Phys. Fluids 11 (1999) 1699.
[43] A. Tyliszczak, J. Comput. Phys. 276 (2014) 438–467.
[44] A. Tyliszczak, A. Wawrzak, K. Wawrzak, Combust. Theory Model. 27 (2023) 244–266.
[45] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, Int. J. Numer. Methods Fluids 72 (2013) 811–845.
[46] Q. Wang, M. Ihme, Combust. Flame 176 (2017) 125–142.
[47] Q. Wang, X. Zhao, M. Ihme, Combust. Flame (2019) 89–100.
[48] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series, The MIT Press, Cambridge, MA, 1949.
[49] D.C. Wilcox, Turbulence Modeling for CFD, DCW Industries, 1993.
[50] P.H. van Cittert, Zum Einfluss der Spaltbreite auf die Intensitätsverteilung in Spektrallinien. II, Z. Phys. 69 (1931) 298–308.
[51] F. van der Bos, J.J.W. van der Vegt, B.J. Geurts, Comput. Methods Appl. Mech. Eng. 196 (2007) 2863–2875.
[52] F. van der Bos, B.J. Geurts, Comput. Methods Appl. Mech. Eng. 199 (2010) 903–915.
[53] R.W.C.P. Verstappen, A.E.P. Veldman, J. Comput. Phys. 187 (2003) 343–368.
[54] P.X. Yu, Z.F. Tian, H. Zhang, Comput. Math. Appl. 73 (2017) 1461–1484.
[55] P. Sagaut, V.K. Suman, P. Sundaram, M.K. Rajpoot, Y.G. Bhumkar, S. Sengupta, A. Sengupta, T.K. Sengupta, Comput. Fluids 261 (2023) 105915.
[56] M.H. Potter, C.B. Morrey, Differentiation under the Integral Sign - Intermediate Calculus, Springer, 1985.
[57] T.M. Apostel, Calculus, Wiley, 1967.
[58] B.J. Geurts, J. Fröhlich, Phys. Fluids 14 (2002) L41–L44.
[59] J. Meyers, B.J. Geurts, M. Baelmans, Phys. Fluids 15 (2003) 2740–2755.
[60] A.W. Vreman, B.J. Geurts, J.G.M. Kuerten, Int. J. Numer. Methods Fluids 22 (1996) 297–311.
[61] J. Meyers, P. Sagaut, B.J. Geurts, Phys. Fluids 18 (2006) 095103.
[62] B.J. Geurts, Direct and Large-Eddy Simulation, Walter de Gruyter GmbH & Co KG, 2022.
[63] P. Cifani, M. Viviani, E. Luesink, K. Modin, B.J. Geurts, Phys. Rev. Fluids 7 (2022) L082601.
[64] M. Oberlack, J. Fluid Mech. 427 (2001) 299–328.
[65] B.J. Geurts, J. Turbul. 1 (2001) 017.
[66] B.J. Geurts, J. Turbul. 7 (2006) N55.
[67] F. van der Bos, B.J. Geurts, Comput. Methods Appl. Mech. Eng. 199 (2010) 903–915.
[68] C. Bogey, C. Bailly, J. Comput. Phys. 194 (2004) 194–214.