SIMULATION OF TURBULENT MIXING USING DECONVOLVED DISCRETIZATION

Bernard J. Geurts
Applied Mathematics – RAM-3MS
University of Twente, P.O. Box 217
7500 AE Enschede, The Netherlands
b.j.geurts@utwente.nl

Lena Caban
Department of Thermal Machinery
Czestochowa University of Technology
al. Armii Krajowej 21, 42-201 Czestochowa, Poland
lena.caban@pcz.pl

Steffen Stolz
Applied Mathematics – RAM-3MS guest
University of Twente, P.O. Box 217
7500 AE Enschede, The Netherlands
steffen.stolz@gmail.com

Andrzej Boguslawski
Department of Thermal Machinery
Czestochowa University of Technology
al. Armii Krajowej 21, 42-201 Czestochowa, Poland
andrzej.boguslawski@pcz.pl

Artur Tyliszczak
Department of Thermal Machinery
Czestochowa University of Technology
al. Armii Krajowej 21, 42-201 Czestochowa, Poland
artur.tyliszczak@pcz.pl

ABSTRACT

Starting from a basic low-order spatial discretisation for the Navier-Stokes equations, we construct new computational methods by approximately inverting the filter induced by the basic discretisation. In this way, the accuracy of a legacy code, used for comprehensively modelling complex flow phenomena, can be improved largely non-intrusively, i.e., without the need to restructure or re-develop the original simulation platform entirely. The key consideration is that any spatial discretisation of the derivative of a solution can be expressed as the exact differentiation of a corresponding 'filtered' solution. By approximately deconvolving the induced filter of the basic numerical discretisation, an augmented method can be obtained, in principle of arbitrary order. This construction is illustrated for a second-order finite difference (FD2) discretisation scheme. 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 induced filter. This corresponds to a 'sharpening' of numerical solution features before the application of the basic FD2 method. We develop the approach for general non-periodic domains using Lagrange interpolation and illustrate the improved accuracy that may be obtained by deconvolution for the well-known dipole-wall interaction flow.

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 (van der Bos, 2007; Verstappen, 2003), or, achieving high formal order of accuracy (Ashcroft, 2003; Caban, 2022; Sengupta, 2016), are two important strategies.

In this paper, we develop an approximate deconvolution of the low-pass filter implied by the second-order central finite difference discretisation (FD2) of the derivative. This enables compensation for the implied filter and achieves a higher-order deconvolved method for the derivative. The implied filter is unavoidably connected to the chosen FD2 method, and will hence reduce the high-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) (Boguslawski, 2024). The main results in this paper are (i) the development of the ADD framework, (ii) its application to the sharpening of the FD2 method for general non-periodic domains using Lagrange interpolation, and (iii) an example based on the dipole-wall interaction flow. 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 an increase in the approximate filter inversion order yields results that systematically converge to high-resolution direct numerical simulation results, even for flows displaying high gradients near solid walls.

Inverse filtering is a general strategy that can be used to recreate some of the small scales in a given signal (Geurts, 2006). A well-known method is through deconvolution. In turbulent flow simulation, filter inversion has received ample attention for modeling of the sub-filter stress tensor in LES. Geurts (Geurts, 1997) developed an approximate higher-order polynomial method for the direct inversion of the top-hat filter and proposed a generalized mixed similarity model. Stolz and Adams (Stolz, 1999) proposed an alternative inversion method, the Approximate Deconvolution Model (ADM), based on repeated filtering, following the van Cittert (van Cittert, 1931) inversion concept. The ADM approach was recently applied by Caban et al. (Caban, 2025) for modelling the sub-grid species fluctuations in LES of a reactive flow.

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. In fact, numerical differentiation based on central differencing can be interpreted as the exact derivative of the function filtered with the top-hat filter (Wilcox, 1993). The implied filter characteristic of a given spatial discretisation was generalized by Geurts and Van der Bos (Geurts, 2005), who showed that finite differencing of an arbitrary order implies a filter composed of a series of local top-hat filters. In fact, in the ADD approach, 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 (Geurts, 1997) of L. Basically, this approximate removal of the implied filter corresponds to

$$\begin{aligned} D_x(u_i) &\equiv \delta_x\Big(L_N^{-1}(u(x_i))\Big) \\ &= \partial_x\Big(L(L_N^{-1}(u(x_i))\Big) \approx \partial_x(u(x_i)) \end{aligned} \tag{1}$$

Hence, to arrive at a higher-order method $D_x(u) \equiv \delta_x(L_N^{-1}(u(x)))$ 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' (Layton, 2011) by systematic insertion of the approximate inverse of the implied filter of the given numerical differentiation.

The organization of this paper is as follows. In Section 2 we present the foundation that underpins the sharpening of finite differencing methods. The approximate van Cittert (van Cittert, 1931) inversion of the filter implied by a specific numerical discretisation is considered in Section 3. Section 4 presents the Lagrange implementation of ADD for non-periodic domains and an application to dipole-wall interaction flow. 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 central finite differencing discretisation. We will establish that a numerical discretisation of a derivative is equivalent to applying the analytic derivative operator to a corresponding filtered solution (Geurts, 2005). 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.

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. Using Gauss's theorem, this discrete approximation may be rewritten as

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

where 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 (Geurts, 1997)

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

which conceptually converges to the exact inverse with increasing N. Various methods to construct the approximate inversion L_N^{-1} are available in literature, e.g., (Geurts, 1997; Stolz, 1999).

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). For this example of sharpening the well-known central discretisation (2) we 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 dimension for an arbitrary finite differencing method on a uniform grid. The numerical derivative $\delta_x(u(x))$ may be written as

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

in terms of a continuous dependence on x. 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.

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 an average of skewed top-hat filters (Geurts, 2005) 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

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

The implied filter L in (8) depends on the stencil $[-n, m]$ and the grid-spacing h of the finite differencing.

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 according to (5), 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

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

with explicit remainder written in terms of a smooth function $\zeta(x, x_0)$ taking values between x and x_0. Correspondingly, using (5), we can take the second step and have

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

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 $u(x)$. More practically, for N-th order inverse designed according to (5), we verify $(L \circ L_N^{-1} - Id)S = 0$ for any polynomial S 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,

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

where use was made of the facts that (i) ∂_x commutes with $L \circ L_N^{-1} - Id$ as we work with a single fixed length-scale parameter h, (ii) $\zeta(x_0, x_0) = x_0$ and, (iii) the function ζ can be expressed as

$$\zeta(x, x_0) = \zeta(x_0, x_0) + \partial_x \zeta(x_0, x_0)(x - x_0) + \ldots \tag{13}$$

This enables to explicitly identify the remainder term in (12) of order N as proposed above and completes the proof.

3 Van Cittert inversion

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 (Geurts, 1997), also van Cittert deconvolution is directly applicable, as was pioneered for turbulence in (Stolz, 1999). 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 (van Cittert, 1931) to approximate the inverse of a filter is based on the formal equivalence

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

by analogy with geometric series. Inspired by this we define

$$L_N^{-1} = \sum_{k=0}^{N} (I - L)^k \tag{15}$$

Some low order examples are

$$N = 0: \quad L_0^{-1} = I \tag{16}$$

$$N = 1: \quad L_1^{-1} = 2I - L \tag{17}$$

$$N = 2: \quad L_2^{-1} = 3I - 3L + L^2 \tag{18}$$

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

$$N = 0: \quad D_{x,0}(u) = \delta_x u \tag{19}$$

$$N = 1: \quad D_{x,1}(u) = 2\delta_x u - \delta_x(L(u)) \tag{20}$$

$$N = 2: \quad D_{x,2}(u) = 3\delta_x u - 3\delta_x(L(u)) + \delta_x(L(L(u))) \tag{21}$$

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

The deconvolution 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

$$L(x^k) = \frac{1}{2(k+1)h}\left((x+h)^{k+1} - (x-h)^{k+1} \right) \quad ; \ k = 0, 1, 2, \ldots \tag{22}$$

We readily infer $L(1) = 1$, $L(x) = x$ and $L(x^2) = x^2 + h^2/3$, and subsequently verify that $L \circ L_N^{-1}(x^k) = x^k$ for $k = 0,1,2$. This verifies that the van Cittert method generates N-th order approximate inverses satisfying requirement (5) for these values of N. Higher values of N can likewise be analysed - this will not be pursued here.

The approximate inverses generated with the van Cittert method can be used to define formally 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. We may derive analytically

$$\begin{aligned} L(u)(x) &= \int_{x-h}^{x+h} d\xi \; \frac{u(\xi)}{2h} = \frac{1}{2h}\int_{x-h}^{x+h} d\xi \; \sin(k\xi) \\ &= \Big(\frac{\sin(kh)}{kh}\Big)\sin(kx) \end{aligned} \tag{23}$$

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

$$\begin{aligned} \delta_x(\sin(kx)) &= \frac{1}{2h}\Big(\sin(k(x+h)) - \sin(k(x-h))\Big) \\ &= \Big(\frac{\sin(kh)}{h}\Big)\cos(kx) = G(kh)\{k\cos(kx)\} \end{aligned} \tag{24}$$

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

$$D_{x,0}(\sin(kx)) = \delta_x(\sin(kx)) = \Big(\frac{\sin(kh)}{kh}\Big)\{k\cos(kx)\} \tag{25}$$

This also establishes the order of this discretisation since

$$D_{x,0}(\sin(kx)) = \Big(1 - \frac{1}{3!}(kh)^2 + \frac{1}{5!}(kh)^4 - \ldots\Big)\{k\cos(kx)\} \tag{26}$$

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

$$\begin{aligned} D_{x,1}(\sin(kx)) &= \delta_x(L_1^{-1}(\sin(kx))) \\ &= \delta_x(2\sin(kx) - G(kh)\sin(kx)) \\ &= 2\delta_x\sin(kx) - G(kh)\delta_x\sin(kx) \\ &= \Big(2 - G(kh)\Big)\Big(G(kh)\Big)\{k\cos(kx)\} \end{aligned} \tag{27}$$

and in general we obtain to leading order

$$\begin{aligned} D_{x,N}(\sin(kx)) &= \delta_x(L_1^{-1}(\sin(kx))) \\ &= \partial_x(L \circ L_N^{-1}(\sin(kx)) \\ &= \partial_x((Id - (Id - L)^{N+1})(\sin(kx)) \end{aligned} \tag{28}$$

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

$$\begin{aligned} D_{x,1}(\sin(kx)) &= \Big(1 - \frac{1}{3!3!}(kh)^4 + \frac{2}{3!5!}(kh)^6 - \ldots\Big) \\ &\times \{k\cos(kx)\} \end{aligned} \tag{29}$$

Likewise, $D_{x,2}$ can be proven to yield sixth-order accuracy. We may generalize this as

$$\begin{aligned} D_{x,N}(\sin(kx)) &= \Big(1 - \Big(\frac{1}{3!}(kh)^2\Big)^{N+1} + \ldots\Big) \\ &\times \{k\cos(kx)\} \end{aligned} \tag{30}$$

This section considered the exact evaluation of the involved filters and their approximate inverses and showed that deconvolution indeed may upgrade a lower-order discretisation into a method with high-order accurate numerical differentiation. In the next section, we present ADD for non-periodic domains and illustrate its enhanced accuracy in dipole-wall interaction flows.

4 Deconvolved finite differencing in non-periodic domains

In this Section, we first develop Lagrange interpolation to apply the general framework of deconvolved discretization to domains with general, non-periodic boundary conditions (Subsection 4.1). Then, in Subsection 4.2, we illustrate the improved accuracy at coarse resolutions achieved by increased deconvolution order. We illustrate this using dipole-wall interaction flow.

4.1 Lagrange interpolation approach

For periodic domains, the same discretisation stencil can be applied at each computational node, making the induced filter spatially invariant. In such cases, the inverse filter can be efficiently constructed using either Fourier-based techniques or an iterative van Cittert deconvolution (Boguslawski, 2024). However, in non-periodic domains, the differentiation stencil changes near boundaries, making the induced filter spatially dependent and requiring a different treatment.

Consider a uniform 1D grid on the interval $[0, L_d]$ with K nodes $x_i = h(i-1)$, $i = 1,..,K$, and mesh spacing $h = L_d/(K-1)$. A general finite difference approximation of the first-order derivative of a function f can be written as (7). For interior nodes sufficiently far from the boundaries, the derivative is evaluated using a fixed stencil $[-n, m]$, where n and m denote the number of nodes to the left and right of x_i, respectively. Near the boundaries, the stencil is locally adjusted through changes in the lower and upper boundaries in the summation, i.e., replacing n and m with n_b and m_b according to

$$\begin{aligned} &\text{for } i = 1,\ldots,n-1 : \quad n_b = i-1,\; m_b = m+n+1-i \\ &\text{for } i = n,\ldots,K-m : \quad n_b = n,\; m_b = m \\ &\text{for } i = K+1-m,\ldots,K : \quad n_b = K+m+n-i,\; m_b = K-i \end{aligned} \tag{31}$$

This construction avoids reaching beyond the computational domain while preserving the total stencil width and formal order of accuracy throughout the domain.

Following Geurts and van der Bos (Geurts, 2005), to evaluate the induced filter in the interior as well as near the boundaries, the integral expression in (4) is approximated using local Lagrange interpolation polynomials, yielding

$$L^{[N^{\mathscr{L}}, M^{\mathscr{L}}]}(u)(x_i) = \frac{1}{h}\sum_{j=-n_b+1}^{m_b} b_j \int_{x_i+(j-1)h}^{x_i+jh} d\eta\, \mathscr{L}(\eta) \tag{32}$$

where $\mathscr{L}(\eta)$ denotes the local Lagrange polynomial approximating f on an interpolation stencil $[-N^{\mathscr{L}}, M^{\mathscr{L}}]$ containing $N^{\mathscr{L}}$ nodes to the left and $M^{\mathscr{L}}$ nodes to the right of node x_i. Near the boundaries, $N^{\mathscr{L}}$ and $M^{\mathscr{L}}$ are adjusted such that $N^{\mathscr{L}} + M^{\mathscr{L}} =$ const at each node. This construction preserves the formal interpolation order across the entire grid, i.e., including the boundary regions. In the present study, a symmetric interpolation stencil is adopted in the interior of the domain, i.e., $N^{\mathscr{L}} = M^{\mathscr{L}}$. The resulting discrete induced filter is represented in matrix form as $\mathbf{L}^{[N^{\mathscr{L}},M^{\mathscr{L}}]} = [L_{ij}]$ where

$$L_{ij} = \int_{x_i+(j-1)h}^{x_i+jh} d\eta \mathscr{L}(\eta) \tag{33}$$

If $\mathbf{L}^{[N^{\mathscr{L}},M^{\mathscr{L}}]}$ is non-singular, its inverse can be computed directly as

$$\mathbf{L}^{-1}_{N^{\mathscr{L}},M^{\mathscr{L}}} = \left[\mathbf{L}^{[N^{\mathscr{L}},M^{\mathscr{L}}]}\right]^{-1} \tag{34}$$

In this formulation, the quality of the inverse depends on the parameters $N^{\mathscr{L}}$ and $M^{\mathscr{L}}$; as the filter width increases, the true inverse is approximated more closely.

The numerical implementation of the Lagrange interpolation approach in ADD is sketched next. In the present study, the base first derivative operator δ_x is constructed using a second-order FD scheme (FD2) consisting of a central scheme in the interior of the domain and one-sided forward/backward schemes at the boundaries. The corresponding discrete differentiation matrix is denoted as

$$\mathbf{D}_x = [d_{i,j}]/h \tag{35}$$

where the coefficients are given by $d_{1,1} = -\frac{3}{2}, d_{1,2} = 2, d_{1,3} = -\frac{1}{2}$, $d_{i,i-1} = -\frac{1}{2}, d_{i,i} = 0, d_{i,i+1} = \frac{1}{2}$ for $i = 2,...,K-1$, $d_{K,K-2} = \frac{1}{2}, d_{K,K-1} = -2, d_{K,K} = \frac{3}{2}$. Correspondingly, the final sharpened ADD operator becomes

$$\mathbf{D}_s^{(1)} = \mathbf{L}^{-1}_{N^{\mathscr{L}},M^{\mathscr{L}}}\mathbf{D}_x \tag{36}$$

which is used throughout the present study as the first-derivative operator.

4.2 Dipole-wall interaction flow

The dipole-wall interaction problem is considered a demanding benchmark for assessing the proposed ADD formulation in unsteady wall-bounded flows. This case involves strong convection, rapid thin boundary-layer formation, and the formation of small-scale vortical structures during the collision process. As shown by Clercx and Bruneau (Clercx, 2006), accurate numerical simulation of this problem is particularly challenging due to the strong sensitivity of the flow to near-wall vorticity production and subsequent vortex-wall interactions.

The flow is solved employing the vorticity-stream function formulation of the Navier-Stokes equations. The solution domain is a square of size $L_x = L_y = 2$ with no-slip boundary conditions imposed on all walls. The initial condition consists of a dipole formed by a pair of counter-rotating vortices. The initial vorticity distribution is defined as $\omega_{1,2}(x,y,0) = \pm\omega_e(1-(r_{1,2}/r_0)^2)\exp(-(r_{1,2}/r_0)^2)$, where $r_{1,2}$ denotes the distance from the vortex centers located at (x_1,y_1) and (x_2,y_2), i.e., $r_1 = \sqrt{(x-x_1)^2+(y-y_1)^2}$, $r_2 = \sqrt{(x-x_2)^2+(y-y_2)^2}$, and r_0 is the vortex radius. The parameter ω_e denotes the extremum of vorticity at $r_{1,2} = 0$. In the present simulations, the vortex centers are located at $(x_1,y_1) = (1.0, 1.1)$ and $(x_2,y_2) = (1.0, 0.9)$, with $r_0 = 0.1$. The initial conditions for the velocity components are given by

$$\begin{aligned} u_0(x,y) &= -\frac{1}{2}\omega_e(y-y_1)\exp(-(r_1/r_0)^2) \\ &\quad + \frac{1}{2}\omega_e(y-y_2)\exp(-(r_2/r_0)^2) \\ v_0(x,y) &= +\frac{1}{2}\omega_e(x-x_1)\exp(-(r_1/r_0)^2) \\ &\quad + \frac{1}{2}\omega_e(x-x_2)\exp(-(r_2/r_0)^2) \end{aligned} \tag{37}$$

The vorticity amplitude $\omega_e \approx 300$ is adjusted such that the initial kinetic energy

$$E(t) = \frac{1}{2}\int_0^2\int_0^2 (u(x,y,t)^2 + v(x,y,t)^2)\mathrm{d}x\mathrm{d}y \tag{38}$$

is equal to $E(0) = 2$. This initialization ensures that both velocity and vorticity remain negligible near the boundaries at $t = 0$, thereby avoiding artificial boundary layer generation (Clercx, 2006).

The computations were performed for the Reynolds number Re $= 625$. The Chebyshev pseudo-spectral (PS) method was used to generate high-fidelity reference solutions. For Re $= 625$, the PS solution was obtained on a non-uniform grid with $K = 257$ Gauss-Lobatto nodes, while the FD2 and ADD methods were tested on uniform grids with $K = 257$ and $K = 513$. We note that FD2/ADD was applied to the convection term only, whereas the diffusion term (2nd order derivative) was discretized using the eighth-order compact difference method. Time integration was performed using a fourth-order Runge-Kutta method. For the PS reference simulations, the time step was set to $\Delta t = 2\times10^{-6}$. All remaining FD2 and ADD simulations, regardless of the grid resolution, were performed with a time step of $\Delta t = 1\times10^{-5}$.

The numerical results are assessed through the temporal evolution of the enstrophy $\Omega(t)$ defined over the entire computational domain as

$$\Omega(t) = \frac{1}{2}\int_0^2\int_0^2 \omega(x,y,t)^2\mathrm{d}x\mathrm{d}y \tag{39}$$

The evolving vorticity field is shown in Figure 1. We clearly observe an intricate vorticity pattern well separated from the walls at early times, which vigorously collide with the walls on the right at $t \approx 0.4$, and also with the walls on the left at $t \approx 1$. Throughout, the initial symmetries are retained during the evolution and very fine structures are formed in the boundary layers.

In Figure 2, the evolution of the enstrophy Ω is shown, comparing the reference PS method with FD2 and its ADD-enhanced versions for different $N^{\mathscr{L}}$. We notice that the ADD results display a significant improvement over the baseline FD2 approach for both computational grids ($K = 257$ and $K = 513$). At the lower resolution ($K = 257$), the differences between solutions obtained for different $N^{\mathscr{L}}$ values are more

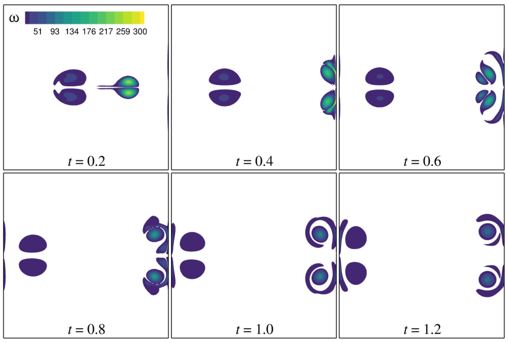

Figure 1. Vorticity contours of the dipole-wall interaction at selected time instances. The results obtained using the PS method on the non-uniform grid with $K = 257$.

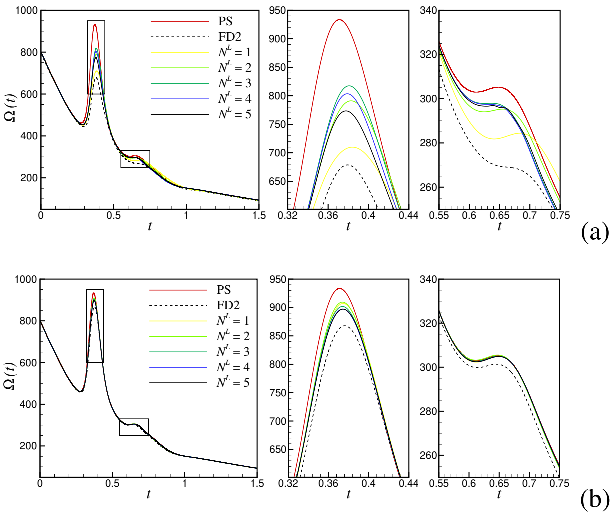

Figure 2. Time evolution of enstrophy for the dipole-wall interaction. The results obtained using the FD2 and ADD methods on grids with $K = 257$ (a) and $K = 513$ (b).

noticeable but do not exhibit a clear monotonic trend. At the higher resolution ($K = 513$), all ADD variants yield very similar results (likely due to time-integration errors) and remain in close agreement with the reference solution. We also studied this flow at higher Reynolds numbers and observed similar behaviour, but with even more refined and slender vortical features in the flow.

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 baseline finite differencing method and correct for these as an integral part of the new method. This is referred to as ‘sharpening’ in which the implied filter associated with the baseline differencing method is approximately inverted.

We showed that theoretically, N-th order discretization can be achieved from any baseline method. For convenience, we focused on the second-order accurate central discretization as baseline method and established fourth- and sixth-order deconvoluted discretization. We illustrated that an approximation with van Cittert inversion can be considered to realize practically realisable implementations.

An extension to non-periodic boundary conditions was achieved using Lagrange interpolation on local grids with a stencil $[-N^{\mathscr{L}}, M^{\mathscr{L}}]$. With increasing $N^{\mathscr{L}}$, $M^{\mathscr{L}}$, the approximate inversion of the induced filter is improved. Clear convergence was found for dipole-wall interaction flow.

Acknowlegments. This work was supported by the National Science Centre, Poland (Grant no. 2024/53/B/ST8/02908) and statutory funds of the Czestochowa University of Technology BS/PB 1-100-301/2026/P. The computations were performed using the high-performance infrastructure PLGrid ACK Cyfronet AGH and PCSS (Poland), and SURF (The Netherlands) under computational grants no. PLG/2025/018864, PL0356-01, and NWO-2024.0016 (Snellius supercomputer) - Multiscale Modeling and Simulation.

REFERENCES

Ashcroft, G. & Zhang, X. 2003 Optimized prefactored compact schemes. *Journal of Computational Physics* **190**, 459–477.

Boguslawski, A., Tyliszczak A. & Geurts B. J. 2024 Approximate deconvolution discretisation. *Computers & Mathematics with Applications* **154**, 175–198.

van der Bos, F., van der Vegt J. J. W. & Geurts B. J. 2007 A multi-scale formulation for compressible turbulent flows suitable for general variational discretization techniques. *Computer Methods in Applied Mechannics and Engineering* **196**, 149–167.

Caban, L. & Tyliszczak, A. 2022 High-order compact difference schemes on wide computational stencils with a spectral-like accuracy. *Computers and Mathematics with Applications* **108**, 123–140.

Caban, L., Tyliszczak A. Geurts B. J. & Domaradzki J. A. 2025 Assessment of LES-ADM Accuracy for Modelling of Auto-Ignition and Flame Propagation in a Temporally-Evolving Nitrogen-Diluted Hydrogen Jet. *Flow, Turbulence and Combustion* **115**, 303–345.

van Cittert, P. H. 1931 Zum einfluss der spaltbreite auf die intensitätsverteilung in spektrallinien. ii. *Z. Physik* **69**, 298–308.

Clercx, H. J. H. & Bruneau, C.-H. 2006 The normal and oblique collision of a dipole with a no-slip boundary. *Computers & Fluids* **35**, 245–279.

Geurts, B. J. 1997 Inverse modelling for large-eddy simulation. *Physics of Fluids* **9** (12), 3585–3587.

Geurts, B. J. 2006 Interacting errors in large-eddy simulation: a review of recent developments. *Journal of Turbulence* **N55**, 1–25.

Geurts, B. J. & van der Bos, F. 2005 A new approach to subgrid surface tension for les of two-phase flows. *Physics of Fluids* **17**, 125103–12.

Layton, W. 2011 Approximate deconvolution discretization of the navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering* **200**, 3183–3199.

Sengupta, T. K. & Sengupta, A. 2016 Non-equilibrium thermodynamics of rayleigh-taylor instability. *Journal of Computational Physics* **310**, 1–25.

Stolz, S. & Adams, N. A. 1999 An approximate deconvolution procedure for large-eddy simulation. *Physics of Fluids* **1**, 1699–1701.

Verstappen, R. W. C. P. & Veldman, A. E. P. 2003 Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics* **187**, 343–368.

Wilcox, D. C. 1993 *Turbulence Modeling for CFD*. DCW Industries.