An optimisation framework for the development of explicit discrete forward and inverse filters

Z. Nikolaou, L. Vervisch, Pascale Domingo

An optimisation framework for the development of explicit discrete forward and inverse filters.

Z. Nikolaou[a], L. Vervisch[a], P. Domingo[a]

[a] *CORIA-CNRS, Normandie Université, INSA de Rouen Normandie, France.*

Abstract

Discrete filters are used in numerous digital signal processing applications and numerical simulations, for anti-aliasing, de-noising, and post-processing. Our specific interest is for application in large-eddy simulations. In this work, we investigate analytically the reconstruction properties of different filters, and how their discrete approximations using different rules affect the convergence and accuracy of the reconstructed signal. Following this analysis, a constrained and adaptive optimisation framework is proposed for the automated calculation of explicit forward but also direct-inverse discrete filter coefficients for a given filter transfer function. The optimised forward filters are shown to perform well with stable reconstruction using classic van Cittert iterations. The optimised direct-inverse filters eliminate the need to apply any iterations thereby substantially reducing the computational cost required for reconstruction which is one of the main challenges associated with deconvolution-based modelling in large-eddy simulations.

Email address: zacharias.nikolaou@insa-rouen.fr (Z. Nikolaou)

1. Introduction

In explicit Large Eddy Simulation (LES), only the largest scales of motion are resolved on the computational mesh by filtering the governing equations over a spatial region $\Omega(\mathbf{x})$ using a filter G [1],

$$G \circledast u \coloneqq \int_{\Omega(\mathbf{x})} G(\mathbf{x}-\mathbf{s};\chi)u(\mathbf{s},t)d\mathbf{s} = \bar{u}(\mathbf{x},t) \tag{1}$$

which, depending on the filter, may substantially damp or completely remove all wavenumbers above the filter cut-off wavenumber k_f. The filter cut-off wavenumber and amount of damping are controlled by the filter parameter χ. Popular filters include the Gaussian, Helmholtz (elliptic differential), Box, Implicit (Padé) [2], and the spectral sharp cut-off filter. The filtering process introduces unclosed terms in the LES equations (scalar flux terms, source terms etc.) which require modelling in order to produce a closed set of equations [3, 4]. This information loss includes contributions from two components: (a) Sub-Grid (SG) scales with wavenumbers $k > \pi/h$ where h is the mesh-spacing, and (b) Resolved Filtered (RF) scales with $k \in [0, \pi/h]$. SG scales are always present for any filter while the amount of damping of the RF scales depends on the filter. For instance, a spectral sharp cut-off filter with a cut-off wavenumber of π/h only involves SG scales since RF scales are not damped. A Gaussian filter on the other hand involves both components. In classic LES, the goal of algebraic models such as the Smagorinsky model [5] is to model the effects of SG scales, and by doing so regularise the governing equations by providing sufficient energy dissipation. If the filter is such that component (b) is present, RF scales also require modelling which can be recovered using approximate/iterative deconvolution.

Deconvolution-based modelling has been considered in numerous studies for RF-

scale modelling, and for SG-scale modelling by exploiting information from reconstructed RF scales. Geurts [6] developed analytic expressions for the inverse of the Box filter, and used these to develop generalised scale-similarity models for the Reynolds stress tensor. Kuerten et al. [7] also developed an analytic expression for the inverse of the Box filter which was then used to develop a dynamic stress-tensor model. The velocity-estimation method of Domaratzki and Saiki [8] may also be thought of as an approximate deconvolution method for the SG scales. In the works of Stolz and Adams [9, 10, 11] the van Cittert [12] reconstruction algorithm was employed to deconvolute damped RF scales up to k_f. In the same study, a novel explicit SG-scale model was proposed by introducing a relaxation term in the momentum equation proportional to the difference between the filtered and deconvoluted velocity fields. This approach was successfully used to model decaying turbulence [9], turbulent channel flow at moderate but also high Re numbers [10, 13], and was also extended for shock-capturing [11]. In [14] an analytic expression was derived for the truncated inverse filter, and applied to the 1D Burger's equation in order to derive an equation for the evolution of the reconstructed field. In [15] implicit SG models were developed in a systematic way by noting that averaging and reconstruction with a Box filter in finite-volume formulations, amounts to filtering and deconvolution, and the procedure was later extended in 3D for the Navier-Stokes equations [16]. In a different approach [17], deconvolution was used as a pre-processing step for increasing the accuracy of the subsequent differentiation scheme. In this approach, the function to be differentiated is first filtered using an approximate inverse filter which is implicitly introduced by the application of the discrete differentiation scheme. Along the same lines, in recent work by Boguslawski et al. [18] inverse Wiener filtering was used to invert the discrete filter implied by the numerical differentiation, and by doing so deconvolve the resolved field on the mesh. Deconvolution-based simulation frame-

works have also been adapted for temporal rather than spatial regularisation [19] but also for application in Lattice-Boltzmann methods [20]. Deconvolution-based modelling has also been applied in numerous a priori and a posteriori studies of turbulent and reacting flows [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]. The rationale behind the RF-modelling approach using deconvolution may also be justified by the fact that the quality of a discretisation scheme generally deteriorates for wavenumbers close to the grid cut-off wavenumber [2], and filtering out these wavenumbers may improve the solution. This point was examined by Bull and Jameson [31] a posteriori using LES of turbulent channel flow. Different LES formulations were tested which included a no-model fully implicit approach, as well as classic explicit approaches (using the Smagorinsky model). The explicit filtering and reconstruction of the RF scales approach, on the LES mesh, was shown to improve the predictions in comparison to the rest of the formulations [31].

Irrespective of the modelling approach and context of use, discrete filters lie at the heart of deconvolution-based LES. The filter transfer function determines the damping of the RF scales which in turn dictates the accuracy/efficiency of the reconstruction. Therefore, the filter itself and its discrete approximation can be considered to be model parameters. A preliminary investigation of different filters on the LES solution was conducted by San et al. [32] using 2D and 3D LES of Taylor-Green vortices. Box filters, Implicit filters [2], and Helmholtz filters were considered. The authors found the simulation results to be sensitive to the filter (and their parameters), but also to the number of total iterations N of the van Cittert reconstruction algorithm. In fact, no clear distinction could be made between the different filters, and similar conclusions were reported in a later study for decaying 1D Burger's turbulence [33].

In practice, discrete filters in deconvolution-based LES may be implicit (Padé)

or explicit. Implicit filters generally offer improved accuracy, and are the preferred choice for canonical flows with periodic boundaries. They do require however solving a linear system in order to obtain the filtered values at all grid-points of the mesh. In most applications, the linear systems are tri-diagonal (three-point Implicit filters used) for which efficient algorithms (e.g. Thomas algorithm) may be used. Nevertheless, this can still lead to implementation and parallelisation/communication issues for application in actual large-scale LES especially for complex configurations, and special consideration is required for an effective implementation [34]. Here, we focus on developing explicit filters which are local, and as a result are straightforward to implement and parallelise.

The primary aim of this work is to examine analytically the effect of different filters on the stability/convergence of the iterative reconstruction algorithm which lies at the heart of most deconvolution-based LES frameworks. This involves analysing the convergence properties of the filters in the continuous domain but also the convergence properties of their discrete approximations. We begin by comparing in section 2 some popular filters. In section 3, convergence criteria for the van Cittert reconstruction algorithm are derived, which is one of the most fundamental linear reconstruction algorithms, and the basis for numerous deconvolution-based LES. The results of this section allow us to gain insight into the effect of the discrete filter transfer function on the reconstruction. Following this, we examine in section 4 the effect of different methods/rules for obtaining explicit discrete filters, and in Section 4.4 an optimisation framework is proposed for developing discrete forward but also direct-inverse filters for a given filter transfer function.

2. Common filters

Table 1 shows some of the most popular explicit and implicit filters. For the Gaussian filter $\Delta' = \Delta/\sqrt{c}$, and typically $c = 6$. For the Box filter if $kL/2 > \pi$ the transfer function becomes negative. Also, even though $\hat{G}(0)$ is not defined for this filter $lim_{k\to 0}\hat{G}(k)$=1. For the spectral cut-off filter, k_c is the cut-off frequency above which all wavenumbers are completely removed. Implicit filters are defined using a linear combination of unfiltered and filtered values [2], and the stencils of the filtered/unfiltered values need not necessarily be equal. If the case where both stencils are symmetric and equal, the Implicit filter is defined using,

$$\sum_{l=-M}^{M} a_l \bar{u}_{i+l} = \sum_{l=-M}^{M} g_l u_{i+l}$$

where $\bar{u}$, u are the filtered and un-filtered (original) values of the signal respectively, and a_l, g_l are the corresponding discrete filter coefficients. The transfer function of this filter is given by,

$$\hat{G}(k) = \frac{g_0 + 2\sum_{l=1}^{M} g_l cos(khl)}{a_0 + 2\sum_{l=1}^{M} a_l cos(khl)}$$

For consistency i.e. $\hat{G}(0) = 1$ we have the condition,

$$\sum_{l=-M}^{M} g_l = \sum_{l=-M}^{M} a_l$$

and $g_{-l} = g_l$, $a_{-l} = a_l$ for symmetric filters. The condition $\hat{G}(\pi) = 0$ may also be imposed so that the grid cut-off wavenumber is always damped. A special case used by Stolz and Adams [9] for deconvolution on a 3-point stencil ($M = 1$) is given in Table 1. The parameter a determines the shape of the transfer function, and $\hat{G}(k) \geq 0$ provided $0 \leq |a| \leq 0.5$.

Filter	$G(x)$	$\hat{G}(k)$
Gaussian	$\frac{1}{\Delta'\sqrt{\pi}} e^{-\frac{x^2}{\Delta'^2}}$	$e^{\frac{-\Delta'^2 k^2}{4}}$
Box	$\begin{cases} \frac{1}{L},\ x \in [-L/2, L/2] \\ 0, \text{ otherwise} \end{cases}$	$\frac{2}{kL} sin\left(\frac{kL}{2}\right)$
Helmholtz	$\frac{1}{2\lambda} e^{-\lvert x\rvert/\lambda}$	$\frac{1}{1+\lambda^2 k^2}$
Spectral cut-off	$\frac{k_c}{\pi} \frac{sin(k_c x)}{k_c x}$	$\begin{cases} 1, k \in [-k_c, k_c] \\ 0, \text{ otherwise} \end{cases}$
Implicit	$a\bar{u}_{i-1} + \bar{u}_i + a\bar{u}_{i+1} =$ $\left(\frac{1}{2}+a\right)\left(\frac{1}{2}u_{i-1} + u_i + \frac{1}{2}u_{i+1}\right)$	$\frac{(1/2+a)(1+cos(kh))}{1+2acos(kh)}$

Table 1: Common explicit and implicit filters with their transfer functions.

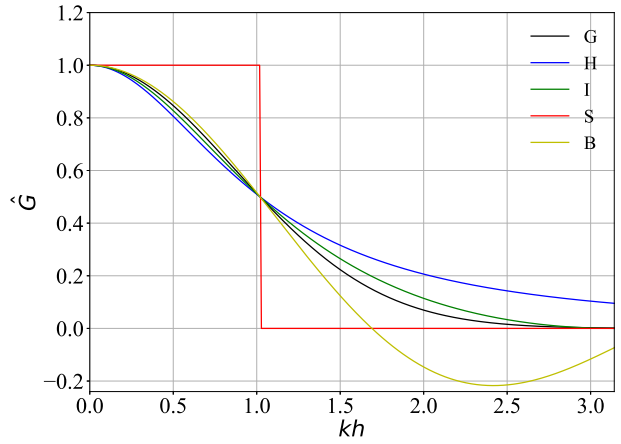


Fig. 1: Filter transfer functions: Gaussian (G), Helmholtz (H), Implicit (I), Sharp cut-off (S), and Box (B). All filters have $k_f h = 1.017 = k_{f,Gaussian} h$ at $\gamma = \Delta/h = 4$.

The Gaussian and Helmholtz filters do not have distinct k_f like the spectral cut-off filter since they are continuous in wavenumber space. As proposed in [10] if we take k_f to correspond to $\hat{G}(k_f) = 0.5$, then for the Gaussian filter $k_f h = \sqrt{24 ln(2)}/\gamma$, where

$\gamma = \Delta/h$ (and taking $c = 6$). For the Helmholtz filter the same k_f can be obtained with $\delta = \lambda/h = 1/(k_f h)$, and for the Implicit filter with $a = -0.5cos(k_f h)$. The corresponding L for the Box filter can be found numerically. Using these values then allows a common basis for comparison. The corresponding filter transfer functions are shown in Fig. 1 for $\gamma = 4$ in which case $k_f h = 1.017$. For $k > k_f$, the Gaussian filter has the least (positive) damping followed by the Implicit/Helmholtz filters, and for $k < k_f$ the situation is reversed. For increasing Δ, L, k_c and λ, k_f reduces and damping increases-for the Implicit filter this occurs for decreasing a. For the Gaussian/Helmholtz filters $\hat{G}(\pi) \neq 0$ while $\hat{G}(\pi) = 0$ always for the Implicit filter by construction. The Box filter is positive for all k provided $L/h \leq 2$, and becomes negative otherwise-as we show later on, this results in unstable reconstruction. Also, from a practical point of view, a reasonable choice would be $L/h = 2$ for the Box filter so that $\hat{G}(k) > 0$. This however fixes k_f which cannot be further reduced. As a result, the Box-filter cannot be compared on an equal basis with the rest of the filters which remain positive for lower k_f values.

3. von Neumann stability analysis of the van Cittert algorithm

The van Cittert algorithm [12] is a fundamental linear reconstruction algorithm which was employed in a number of LES studies [9, 10, 33, 35, 36]. Due to its linearity/simplicity it has also been the subject of numerous theoretical investigations, and the basis for developing simplified analytical models for the unresolved stress tensor [37, 38, 39, 40]. The algorithm reads,

$$\phi^{*N+1}(\mathbf{x}) = \phi^{*N}(\mathbf{x}) + b\left(\bar{\phi}(\mathbf{x}) - G \circledast \phi^{*N}(\mathbf{x})\right) \tag{2}$$

where ϕ^{*N} is the estimate of the reconstructed signal at iteration N, the initial

condition is taken to be $\phi^{*0} = \bar{\phi}$, and b is a constant. In order to investigate the convergence properties of the algorithm a von Neumann stability analysis is conducted. For periodic functions in $x \in [0, L]$, the corresponding Fourier series expansions of the original/filtered signals are,

$$\phi(x) = \sum_{r=-\infty}^{\infty} a_r e^{jk_r x} \qquad \bar{\phi}(x) = \sum_{r=-\infty}^{\infty} a_r e^{jk_r x} \hat{G}(k_r)$$

$$\phi^{*N}(x) = \sum_{r=-\infty}^{\infty} A_r^N e^{jk_r x} \qquad \bar{\phi^*}(x) = \sum_{r=-\infty}^{\infty} A_r^N e^{jk_r x} \hat{G}(k_r)$$

where $k_r = 2\pi r/L$ and,

$$\hat{G}(k_r) = \int_{-\infty}^{\infty} G(s) e^{-jk_r s} ds$$

Inserting the Fourier expansions into Eq. 2 we obtain after some algebraic manipulation a recurrence relation for the evolution of the Fourier coefficients A_r^N, $A_r^{N+1} = A_r^N \left(1 - b\hat{G}(k_r)\right) + b a_r \hat{G}(k_r)$, which is a modified geometric series with the solution $A_r^N = (A_r^0 - a_r)\left(1 - b\hat{G}\right)^N + a_r$. Since $A_r^0 = a_r \hat{G}(k_r)$ (initial condition) it is straightforward to show that,

$$\phi^{*N}(x) = \underbrace{\sum_{r=-\infty}^{\infty} a_r e^{jk_r x}}_{\phi(x)} + \sum_{r=-\infty}^{\infty} \left(1 - b\hat{G}(k_r)\right)^N (\hat{G}(k_r) - 1) a_r e^{jk_r x} \tag{3}$$

where the second term in Eq. 3 is essentially the reconstruction error. An alternative interpretation of Eq. 3 is in terms of the transfer function of the reconstructed field $\hat{Q}^N$,

$$\hat{Q}^N(k_r) = \frac{A_r^N}{a_r} = 1 - \left(1 - b\hat{G}(k_r)\right)^N (1 - \hat{G}(k_r)) \tag{4}$$

where the approximate inverse $\hat{V}^N$ is $\hat{V}^N(k_r) = \hat{Q}^N(k_r)/\hat{G}(k_r)$. For convergence we require,

$$0 < b\hat{G}(k_r) < 2 \tag{5}$$

so $\hat{Q}^N(k_r) \to 1$ for large N and for all $\hat{G}(k_r) \neq 0$. Note that if $\hat{G}(k_r) = 0$ then $\hat{Q}^N(k_r) = 0$ i.e. reconstruction has no effect on these wavenumbers. In the case of a Box filter with $L/h > 2$, its transfer function is no longer positive throughout $[0, \pi]$, and according to Eq. 4 reconstruction is unstable. For a Gaussian filter on the other hand $\hat{G}(k) = e^{-\Delta^2 k^2/24} = e^{-c_r} > 0$, and we have the following cases,

$$b \begin{cases} b = e^{c_r}, & \text{immediate convergence for wavenumber } r \\ 0 < b < e^{c_r}, & \text{non-oscillating reducing convergence for wavenumber } r \\ e^{c_r} < b < 2e^{c_r}, & \text{oscillating reducing convergence for wavenumber } r \end{cases}$$

and similar expressions can be derived for the rest of the filters. Note that equation 5 holds irrespective of $\hat{G}$, and the choice $b = 1$ is a reasonable one to ensure steady convergence.

In the discrete case on a mesh with $x_i{=}ih$, $i \in [0, N_x - 1]$, $N_x h = L$, the algorithm is applied point-wise to obtain $\phi_i^{*N} = \phi^{*N}(ih)$,

$$\phi_i^{*N+1} = \phi_i^{*N} + b(\bar{\phi}_i - G_d \circledast \phi_i^{*N}) \tag{6}$$

where $\phi_i^{*0} = \bar{\phi}_i$, and where $\circledast$ now denotes the discrete filtering operation. The simplest explicit rule we can employ is,

$$G_d \circledast u_i \coloneqq \bar{u}_i = \sum_{l=-M}^{M} g_l u_{i+l} \tag{7}$$

where g_l are the discrete filter coefficients-the different methods for obtaining discrete filters and how these affect reconstruction are discussed in section 4. Inserting the corresponding Fourier expansions for the discrete signals in Eq. 6, and following a similar procedure as in the continuous case we eventually obtain the discrete analog of Eq. 3,

$$\phi_i^{*N} = \underbrace{\sum_{r=0}^{N_x-1} a_r e^{jk_r ih}}_{\phi_i} + \sum_{r=0}^{N_x-1} \left(1 - b\hat{G}_d(k_r)\right)^N \left(\hat{G}_d(k_r) - 1\right) a_r e^{jk_r ih}$$

where $k_r = 2\pi r/(N_x h)$ and,

$$\hat{G}_d(k_r) = \sum_{l=-M}^{M} g_l e^{jk_r hl} = g_0 + 2\sum_{l=1}^{M} g_l cos\left(k_r hl\right) \tag{8}$$

As in the continuous case the function,

$$\hat{Q}_d^N(k_r) = 1 - \left(1 - b\hat{G}_d(k_r)\right)^N \left(1 - \hat{G}_d(k_r)\right) \tag{9}$$

is the discrete approximation at iteration N of the transfer function of the reconstructed field and for convergence Eq. 5 must hold for $\hat{G}_d$. In such a case, and provided $0 < \hat{G}_d < 1$, for a sufficiently large number of iterations N, $\hat{Q}_d^N(k_r) \to 1$. In practice, $\hat{G}(k_r)$ and $\hat{G}_d(k_r)$ are not necessarily equal due to approximation errors in the discrete representation of the filter. Suppose $\hat{G} = \hat{G}_d + e$ where e is an error term and $b = 1$. Inserting this into Eq. 9 and expanding we obtain $\hat{Q}_d^N(k_r) = 1 - \left[\left(1 - \hat{G}(k_r)\right)^{N+1} + (N+1)\left(1 - \hat{G}(k_r)\right)^N (e) + ... + (e)^{N+1}\right]$. Therefore, the reconstructed discrete $\hat{Q}_d^N$ will also contain an error which is equal to the sum of all the terms in the parenthesis starting from the second term. This error depends on N, and on $\hat{G}$.

γ-Gaussian	δ-Helmholtz	a-Implicit	$k_f h$	$k_f h/\pi$	N_G	N_H	N_I
4	0.981	-0.262	1.017	0.324	46	13	26
6	1.471	-0.389	0.677	0.215	47	13	17
8	1.961	-0.436	0.509	0.162	47	13	15
10	2.452	-0.459	0.404	0.129	47	13	14
12	2.942	-0.471	0.336	0.107	47	13	14
14	3.432	-0.479	0.288	0.092	48	13	14
16	3.923	-0.484	0.252	0.080	48	13	14

Table 2: Number of van Cittert iterations N required for each filter having the same initial k_f for recovering all $k \in [0, 2k_f]$.

3.1. Convergence rate

The choice of total iterations N is a free parameter. If N is sufficiently large and the stability condition is satisfied all $k_r h \in [0, \pi]$ are recovered. Depending on the context where reconstruction is employed this may or may not be desirable. In numerical simulations for instance this may not be desirable because wavenumbers close to the grid cut-off are generally not well represented by the numerical scheme [2]. In actual LES, it would be reasonable to chose $\gamma/\delta/a$ such that $k_f \leq k_s$, where k_s is the wavenumber beyond which the accuracy of the numerical scheme deteriorates. For instance, a 2nd-order-accurate centered scheme for the first derivative has $k_s h \simeq 1.0$ [2]. For $\gamma = 4$, $k_f h = 1.017$. Therefore, if a Gaussian filter with $\gamma \geq 4$ is used, all wavenumbers larger than k_s will be damped substantially. In any case, the filter cut-off k_f must be recovered. To recover k_f, we find that for $N = 4$ the error is 0.5^{4+1} i.e. less than 5% irrespective of the filter used. Indeed, $N = 3 - 5$ iterations with an Implicit filter ($a = 0.25$) were used in previous studies [9, 32, 33].

In order to compare the convergence rate between the different filters, the number of iterations required to recover wavenumbers up to $2k_f$ from the same initial

filter cut-off k_f have been calculated. For this exercise, a wavenumber is considered "recovered" if more than 95% of its damping has been recovered. Wavenumbers larger than $2k_f$ may in principle be recovered but for heavier damping the number of required iterations becomes substantially large. The number of iterations for each filter and for different filter parameter values are given in Table 2. Note that all filters have the same initial k_f. For γ =4, both $\hat{G} = \hat{Q}^0$ and $\hat{Q}^N$ for each filter are shown in Fig. 2. Table 2 shows that the Gaussian filter requires the most iterations, and the Helmholtz filter the least since it damps higher wavenumbers less. Table 2 also reveals that the number of iterations for the Gaussian and Helmholtz filters are relatively insensitive to the filter parameter value but this is not the case for the Implicit filter. In addition, the reconstructed transfer function of the Helmholtz filter shows that wavenumbers close to the grid cut-off are substantially amplified which may not be desirable. In terms of the reconstruction computational efficiency, this does not depend only on N, but additionally depends on the stencil size required to represent a filter accurately.

4. Discrete approximation of continuous filters

In order to develop discrete filters Eq. 1 must be approximated numerically where the function values u_i are known only at the distinct points x_i. The simplest explicit rule we can employ is given by Eq. 7 with the corresponding transfer function given by Eq. 8-this is always real for symmetric filters. Ideally, $\hat{G}_d(k_r h) = \hat{G}(k_r h)$ for all $k_r h \in [0, \pi]$, and the half-stencil size M should be such that the filter is well-resolved on the mesh. A straightforward way to quantify this is to use the consistency condition $\int_{-\infty}^{\infty} G(x)dx = 1$. For the Box filter which is compact in space the integration interval is $[-L/2, L/2]$, and $2Mh = L$. For the Gaussian/Helmholtz filters which are not compact we may define the consistency error $e_c = 1.0 - \int_{-x_o}^{x_o} G(x)dx$,

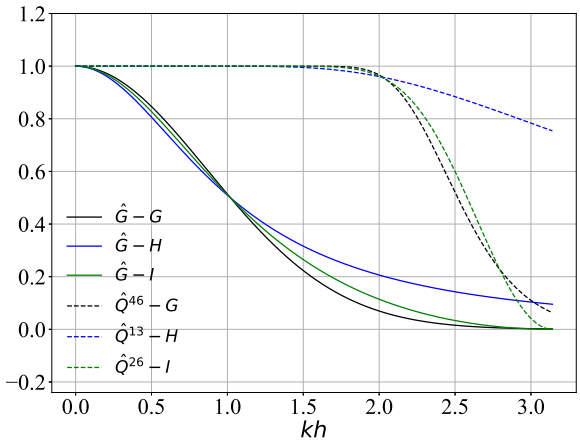


Fig. 2: Forward $\hat{G}$ (=$\hat{Q}^0$) and reconstructed $\hat{Q}^N$ transfer functions for the Gaussian (G), Helmholtz (H), and Implicit (I) filters. The number of iterations N for each filter (Table 2 for $\gamma = 4$) is such that all $k \in [0, 2k_f]$ are recovered. For this case $k_f h \simeq 1$ and all wavenumbers up to about kh =2.0 are recovered.

where $2Mh = 2x_o$ is the symmetric region of integration. The errors for the Gaussian/Helmholtz filters $e_{c,G}/e_{c,H}$ are given by $e_{c,G} = 1 - erf(x_o\sqrt{6}/\Delta)$ and $e_{c,H} = exp(-x_o/\lambda)$ respectively. For a given consistency error the corresponding x_o can be calculated, and the required stencil size M obtained for each filter. Table 3 shows the corresponding x_o for different values of the consistency error.

e_c	x_o/Δ-Gaussian	x_o/λ-Helmholtz
10^{-3}	0.95	6.91
10^{-4}	1.12	9.21
10^{-5}	1.28	11.51
10^{-6}	1.41	13.82

Table 3: Consistency error e_c and required half-integration region x_o for the Gaussian and Helmholtz filters.

Table 3 shows that a good approximation would be $M = int(\gamma x_o/\Delta)$ for the Gaussian filter, and $M = int(\delta x_o/\lambda)$ for the Helmholtz filter. In practice, and keeping efficiency in mind, a consistency error between 10^{-3} and 10^{-4} leads to a choice of $[-1\Delta, 1\Delta]$ for the Gaussian filter, and $[-7\lambda, 7\lambda]$ for the Helmholtz filter. Note that the consistency error of the discrete filter depends on the rule/method we employ. If for the above stencil sizes the discrete filter is accurate enough, the discrete consistency error should match the continuous consistency error as given in Table 3. For the Implicit filter which is not easily recoverable in physical space, we will show in section 4.4 how the stencil size can be determined adaptively using optimisation.

4.1. Newton-Cotes rules.

Newton-Cotes (NC) rules are the natural choice for evaluating directly Eq. 1. One approach is to apply a full rule on the $(2M + 1)$-point stencil of the filter (fit a $2M$-degree Lagrange polynomial), and another is to apply a lower-order rule recursively i.e. use a composite rule. Here, we consider full rules as well as the 2-point composite trapezium rule (NC2), and the 3-point composite Simpson's rule (NC3). Table 4 lists the NC integration coefficients (given by $c_l = b_l h/f$) up to a 9-point stencil. A composite NC2 rule can be perfectly applied for any stencil while a composite NC3 rule can be perfectly applied for any odd-numbered stencil. For instance, with the NC2 rule and a 5-point stencil, the integration coefficients are $h \cdot [1/2, 1, 1, 1, 1/2]$ since on every point excluding the boundaries the $1/2$ integration coefficients are added together. The filter coefficients, call these z_l, are then given by $h \cdot [1/2 \cdot G(-2h), 1 \cdot G(-h), 1 \cdot G(0), 1 \cdot G(h), 1/2 \cdot G(2h)]$ from which the discrete transfer function can be calculated using,

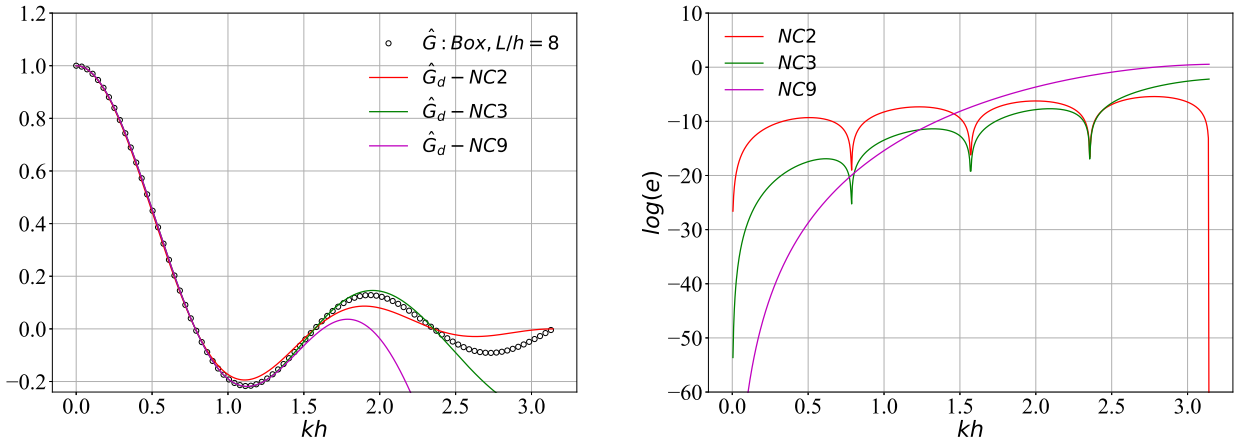
Points	f	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
2	2	1	1							
3	3	1	4	1						
5	45	14	64	24	64	14				
7	140	41	216	27	272	27	216	41		
9	14175	3956	23552	-3712	41984	-18160	41984	-3712	23552	3956

Table 4: Integration coefficients c_l for Newton-Cotes rules are given by $c_l = b_l h/f$.

$$\hat{G}_d(k_r h) = \left(z_0 + 2 \sum_{l=1}^{M-1} z_l cos(k_r h l) \right)$$

where z_l are the coefficients on each stencil point calculated as in the example above. For other rules, the same process can be applied. In order to examine the response of the different NC rules, we consider a case where filtering amounts to integration. This corresponds to filtering using a Box filter on a $(2M+1)$-point stencil with $L = 2Mh$. In this case, filtering is the same as integrating u over L scaled by $1/(2Mh)$. For the example here, we choose $M = 4$ i.e. a 9-point stencil. On the 9-point stencil we apply a 9-point NC rule, but also composite NC2/NC3 rules. Fig. 3(a) shows the actual transfer function, and the discrete transfer functions using the NC2/NC3/NC9 rules. The corresponding squared error, $e(kh) = \left(\hat{G}_d(kh) - G(kh)\right)^2$, is shown in Fig. 3(b). The results in Fig. 3 indicate that higher-order rules may not necessarily be beneficial. The NC9 rule reduces the error but only for lower wavenumbers whereas for larger wavenumbers the error increases beyond that of the composite NC2/NC3 rules. Also, at $kh = \pi$ only the NC2 rule is consistent. Therefore, if the signal contains mostly low wavenumbers, a high-order rule would be appropriate but if it mostly contains higher-wavenumbers a higher-order rule would yield poorer results. In the sections which follow, we show that for non-constant filters, higher-order rules

lead to poorer results even for lower wavenumbers.



(a) (b)

Fig. 3: (a) Actual ($\hat{G}$) and discrete ($\hat{G}_d$) filter transfer functions obtained using a 9-point Newton-Cotes rule (NC9) as well as composite NC2/NC3 rules on a 9-point stencil for the Box-filter with $L/h = 8$, and (b) logarithm of squared error $e(kh) = \left(\hat{G}_d(kh) - G(kh)\right)^2$ against wavenumber.

4.2. Taylor expansion

An alternative approach to develop discrete filters is using Taylor-series expansions which results in a differential expression for the filter [41]. Starting from the 1D filtering definition,

$$\bar{u}(x,t) = \int_{-\infty}^{\infty} G(x-s)u(s,t)ds = \int_{-\infty}^{\infty} G(s)u(x-s,t)ds \tag{10}$$

and expanding u in Taylor series around $s = x$ we obtain (from now on we will drop the time dependency of u which will be implied),

$$u(s) = \sum_{r=0}^{\infty} \frac{(s-x)^r}{r!} u^r(s)|s = x \tag{11}$$

where $u^r(s) = \partial^r u / \partial s^r$ is the r_{th}. Inserting the Taylor series expression into Eq. 10 we obtain,

$$\bar{u}(x) = \sum_{r=0}^{\infty} \frac{(-1)^r u^r(x)}{r!} \int_{a=-\infty}^{\infty} a^r G(a) da = \sum_{r=0}^{\infty} \frac{(-1)^r u^r(x)}{r!} M_r \tag{12}$$

i.e. the filtered signal is a sum of the derivatives of u whose factors depend on the moments M_r of the filter. In the case of symmetric filters all odd moments disappear leading to,

$$\bar{u}(x) = u(x) + \sum_{r=1}^{\infty} \frac{u^{2r}(x)}{(2r)!} M_{2r} \tag{13}$$

Table 5 shows the corresponding formulas for the filters' moments along with a truncated Taylor expansion.

Filter	M_{2r}	Taylor expansion
Gaussian	$\frac{(2r)!}{r!}\left(\frac{\Delta'}{2}\right)^{2r}$	$\bar{u}(x) = u(x) + \frac{\Delta'^2}{4}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta'^4}{32}\frac{\partial^4 u}{\partial x^4} + \frac{\Delta'^6}{384}\frac{\partial^6 u}{\partial x^6} + \frac{\Delta'^8}{6144}\frac{\partial^8 u}{\partial x^8} + O(\Delta'^{10})$
Box	$\frac{1}{2r+1}\left(\frac{L}{2}\right)^{2r}$	$\bar{u}(x) = u(x) + \frac{L^2}{24}\frac{\partial^2 u}{\partial x^2} + \frac{L^4}{1920}\frac{\partial^4 u}{\partial x^4} + \frac{L^6}{322560}\frac{\partial^6 u}{\partial x^6} + \frac{L^8}{92897280}\frac{\partial^8 u}{\partial x^8} + O(L^{10})$
Helmholtz	$(2r)!\lambda^{2r}$	$\bar{u}(x) = u(x) + \lambda^2\frac{\partial^2 u}{\partial x^2} + \lambda^4\frac{\partial^4 u}{\partial x^4} + \lambda^6\frac{\partial^6 u}{\partial x^6} + \lambda^8\frac{\partial^8 u}{\partial x^8} + O(\lambda^{10})$

Table 5: Filter moments and Taylor expansions

Since the Taylor series expansion includes derivative terms, a straightforward approach to derive discrete filters is to first truncate the series up to a certain order, and then approximate the remaining derivatives using a finite difference rule. In practice, the stencil size imposes a restriction on the maximum number of terms which can be retained. For instance, on a 3-point stencil the 2nd derivative is approximated using an $O(h^2)$ scheme, on a 5-point stencil we can approximate up to the 4th derivative

etc. As a result, for a given stencil size we truncate up to maximum number of derivatives we can approximate. In this approach, the order of the finite difference approximation is highest for the lowest derivative and lowest for the highest derivative retained in the expansion. For instance, for the Gaussian filter and using $c = 6$ we have,

3-point stencil:

$$\bar{u}(x) = u(x) + \frac{\Delta^2}{24}\frac{\partial^2 u}{\partial x^2}$$

where the 2nd derivative finite-difference approximation is $O(h^2)$.

5-point stencil:

$$\bar{u}(x) = u(x) + \frac{\Delta^2}{24}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta^4}{1152}\frac{\partial^4 u}{\partial x^4}$$

where the 2nd derivative approximation is $O(h^4)$, and the 4th derivative approximation is $O(h^2)$ and so on for increasing stencil sizes.

4.3. Relation of Taylor-expansion to Newton-Cotes

The largest degree polynomial we can fit on a $(2M + 1)$-point stencil is a $2M$-degree Lagrange polynomial $p_{2M}(x)$. If this polynomial is integrated over the finite region $[-Mh, Mh]$ we obtain the coefficients for a NC rule. However, the NC rule obtained using this approach i.e. as was done in section 4.1 does not contain any information about the filter G. If this polynomial is used to approximate the function $u(x)$ alone (rather than Gu) everywhere in the domain then we have,

$$\bar{u}_i = \int_{-\infty}^{\infty} G(s)u(x_i - s)ds = \int_{-\infty}^{\infty} G(s)p_{2M}(x_i + s)ds$$

where the $2M$-degree Lagrange polynomial is given by,

$$p_{2M}(x_i + s) = \sum_{r=-M}^{M} u_{i+r} \prod_{j=-M,(i+j)\neq(i+r)}^{j=M} \frac{(s + x_i - x_{i+j})}{(x_{i+r} - x_{i+j})}$$

Inserting into the convolution integral we obtain,

$$\bar{u}_i = \sum_{r=-M}^{M} u_{i+r} \left[\int_{-\infty}^{\infty} G(s) \prod_{j=-M,(i+j)\neq(i+r)}^{j=M} \frac{(s + x_i - x_{i+j})}{(x_{i+r} - x_{i+j})} ds \right]$$

where the term in brackets is the weight g_r for each quadrature point which depends on the filter. For a uniform mesh, the weights g_r only depend on h and after some algebraic manipulation these can be shown to be given by,

$$g_r = \frac{(-1)^{(M-r)}}{h^{2M}(M+r)!(M-r)!} \int_{-\infty}^{-\infty} G(s) \prod_{j=-M,j\neq r}^{j=M} (s - jh)\, ds \tag{14}$$

Since the filter is symmetric, all the odd terms in the polynomial expansion disappear, and the integral reduces to a sum of all the even moments of the filter. In addition, consistency will be satisfied, and this approach thus becomes equivalent to the Taylor-series method described above. The difference is that this approach results in a direct expression for obtaining the filter coefficients (Eq. 14) which alleviates the need to expand the Taylor series and to apply differentiation rules to each resulting term. The coefficients g_r for the Gaussian, Helmholtz and Box filters are given up to a 9-point stencil in the Appendix. In contrast to the NC rules the filter coefficients are functions of the filter parameter values.

4.4. Optimised forward filters

Filters using NC rules are not consistent, and as shown in section 4.1 higher-order NC rules generally have poorer high-wavenumber responses. Filters based on Taylor expansions are consistent with improved responses, however the maximum range of filter parameter values (γ, δ, L/h) for which they work well is limited by

the stencil size. It would also be useful to be able to obtain discrete filter coefficients from knowledge of $\hat{G}$ alone for cases where $\hat{G}$ is given in Fourier space only with no straightforward inverse transform (as is the case of the Implicit filter). This is not straightforward to do with the above approaches. In addition, NC rules and Taylor-expansion do not provide explicit control of the approximation error nor do they ensure that $\hat{G}_d$ remains positive (ensuring reconstruction stability). All of these issues can be addressed using optimisation. The optimisation problem we propose solving is,

$$\arg\min_{g_l} \left(\int_{kh=0}^{\pi} \left(\hat{G}_d(kh; g_l) - \hat{G}(kh; \chi) \right)^2 d(kh) \right) \tag{15}$$

The optimisation constraints are slightly different for the Gaussian/Helmholtz/Implicit filters. For all three filters to ensure consistency ($\hat{G}(0) = 1$) we apply the constraint,

$$g_0 + 2\sum_{l=1}^{M} g_l = 1$$

To ensure $0 \leq \hat{G}_d \leq 1$ for the Gaussian and Helmholtz filters, rather than specifying a fixed lower threshold we found the results to be improved when applying the constraint,

$$\hat{G}(\pi) \leq g_0 + 2\sum_{l=1}^{M} g_l cos(khl) \leq 1 \ \ \forall \ \ kh \in (0, \pi]$$

instead i.e. having a filter-dependent lower-threshold, $\hat{G}(\pi)$, which is the minimum value of the transfer function for the Gaussian and Helmholtz filters. For the Implicit filter we apply instead the constraints,

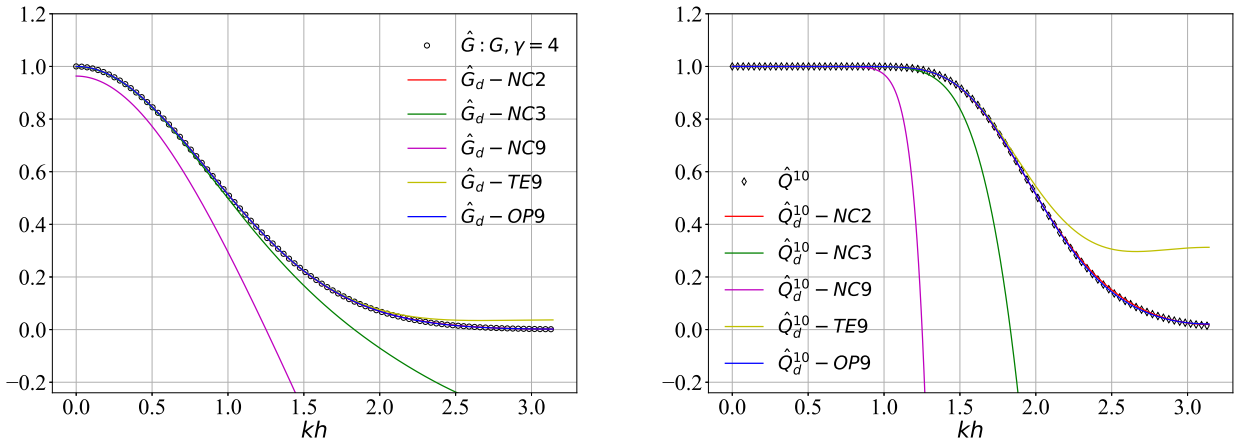
$$g_0 + 2\sum_{l=1}^{M} g_l cos(\pi l) = 1$$

$$\hat{G}(\pi) \leq g_0 + 2\sum_{l=1}^{M} g_l cos(khl) \leq 1 \ \forall \ kh \in (0, \pi)$$

where the first constraint ensures $\hat{G}_d(\pi) = 0$ as should be for the Implicit filter. An optimisation approach for fixed 3-point and 5-point stencils was employed by Sagaut and Grohens [41] to derive discrete filters in the context of test-filtering operations for explicit LES modelling (Gaussian/Box filters). In this work, additional constraints are imposed, and two different methods are proposed for determining the filter-dependent stencil size: (a) using the results of section 4, and (b) adaptively. In method (a), for a given filter parameter value the stencil size is determined using $M = (int(2\gamma) + 1)$ for the Gaussian, and $M = (int(2 \cdot 7\delta) + 1)$ for the Helmholtz filter. In method (b), the Mean Squared Error (MSE) between $\hat{G}_d$ and $\hat{G}$ is specified (essentially the term in parenthesis in Eq. 15), and the stencil size is obtained adaptively until the target error is obtained.

The constrained optimisation problem was implemented in Python along with dedicated libraries, and a Sequential Least Squares Programming (SLSQP) optimiser was used to solve it (the SLSQP optimiser was found to give the optimum results). In the adaptive version of the optimisation routine, the mesh size is increased sequentially and the optimisation problem solved on each new stencil. An error controller is used to check at every iteration whether the target error on the new stencil has been obtained, and whether the optimisation error increases with every iteration in which case the routine terminates.

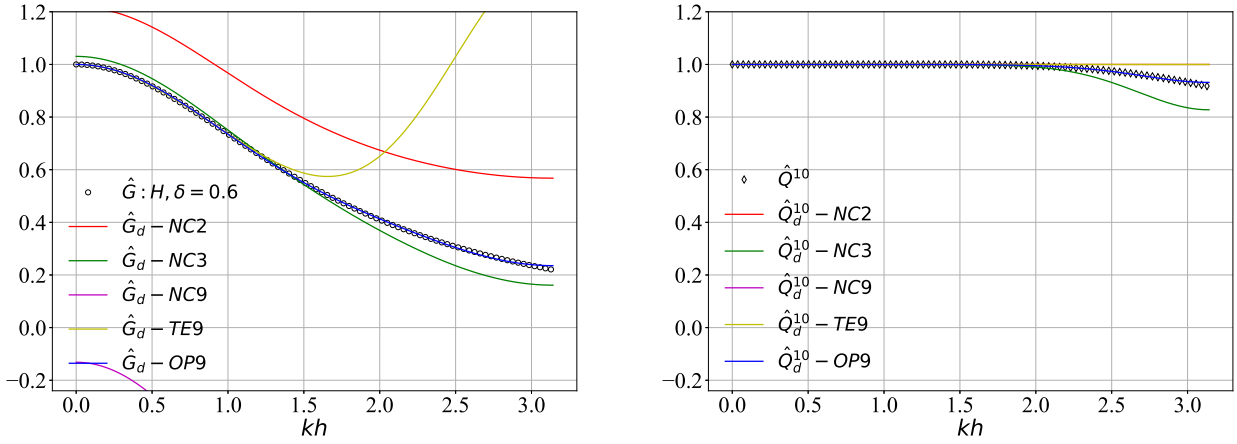
In order to demonstrate the method, optimised coefficients were obtained on a 9-point stencil for a Gaussian filter with $\gamma = 4$, and for a Helmholtz filter with δ =0.6.



(a) (b)

Fig. 4: (a) Actual ($\hat{G}$) and discrete ($\hat{G}_d$) filter transfer functions for the Gaussian filter having $\gamma = 4$ on a 9-point stencil using the different rules: NC2 (trapezium rule), NC3 (Simpson's rule), NC9 (9-point Newton-Cotes), TE9 (9-point Taylor-expansion), OP9 (9-point optimised filter), (b) corresponding actual ($\hat{Q}$) and discrete ($\hat{Q}_d$) reconstructed transfer functions at 10 van Cittert iterations.

These values were chosen according to the rules in section 4. Note however that the discrete consistency error varies substantially between the different rules. Figures 4 (a) and 5 (a) show the actual and discrete filter transfer functions obtained using the different rules for the Gaussian and Helmholtz filters respectively. Figures 4 (b) and 5 (b) on the other hand show the corresponding actual and discrete reconstruted transfer functions at $N = 10$ van Cittert iterations. For the Gaussian filter, the NC2 rule and the optimised filter perform well followed by the filter based on Taylor-expansion with their transfer functions being almost indistinguishable from the actual one. The reconstructed transfer functions of the NC2 rule and of the optimised filter also match closely the actual reconstructed transfer function as one may see from Fig. 4 (b). The reconstrcuted transfer function for the TE filter somewhat amplifies

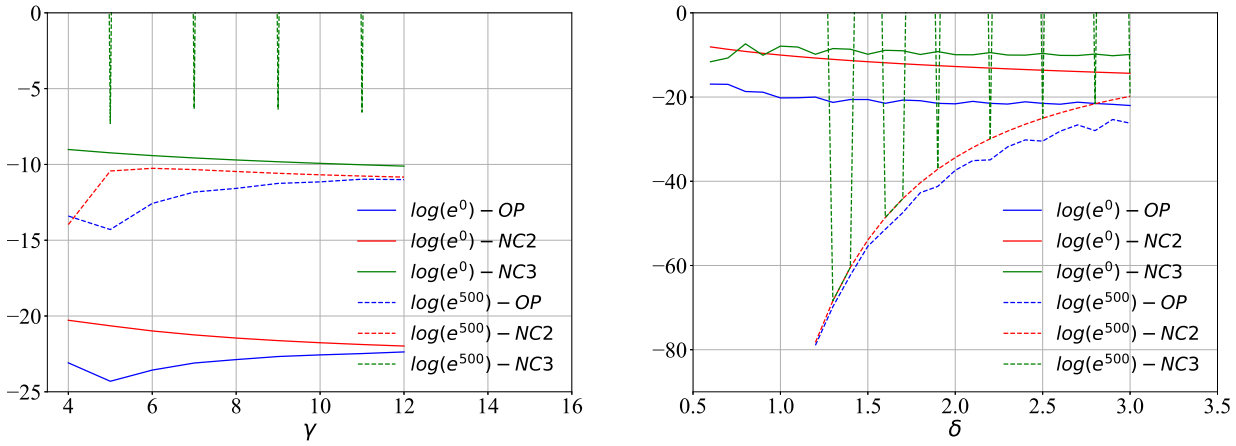


(a) (b)

Fig. 5: (a) Actual ($\hat{G}$) and discrete ($\hat{G}_d$) filter transfer functions for the Helmholtz filter having $\delta = 6$ on a 9-point stencil using the different rules: NC2 (trapezium rule), NC3 (Simpson's rule), NC9 (9-point Newton-Cotes), TE9 (9-point Taylor-expansion), OP9 (9-point optimised filter), (b) corresponding actual ($\hat{Q}$) and discrete ($\hat{Q}_d$) reconstructed transfer functions at 10 van Cittert iterations.

higher wavenumber contributions because of the lower damping obsreved for $\hat{G}_d$ at these wavenumbers. The NC3 rule performs poorly in comparison while the NC9 performs the worst even at lower wavenumbers which is in contrast to the results obtained in section 4 for constant filters. For the Helmholtz filter, the NC3 rule performs better than the NC2 rule which has a larger consistency error, but not as well as the optimised filter. The filter based on Taylor-expansion performs well only for a limited range of wavenumbers while the NC9 rule has the worst performance- $\hat{G}_d(0)$ is negative for this filter because the 8th degree Lagrange polynomial fitted over the 9-point stencil is a poor approximation for this particular choice of δ. This results in unstable reconstruction, and in fact $\hat{Q}_d^{10}(0)$ is so negative that $\hat{Q}_d^{10}$ exceeds

the limits of Fig. 5 (b) for all wavenumbers.



(a) (b)

Fig. 6: Error as as defined in Eq. 16: (a) $e^N(\gamma)$ for the Gaussian filter and (b) $e^N(\delta)$ for the Helmholtz filter. Note that the error for $N = 0$ corresponds to the error for the forward filter.

The above results indicate that the approximation rule is sensitive to $\hat{G}$ whose shape depends on the parameter value. In order to quantify the performance of each rule for different filter parameter values we calculate the error measure,

$$e^N(\chi) = \int_{kh=0}^{\pi} \left(\hat{Q}_d^N(kh; g_l) - \hat{Q}^N(kh; \chi)\right)^2 d(kh) \tag{16}$$

where $\chi = \gamma$ for the Gaussian filter and $\chi = \delta$ for the Helmholtz filter. This was done for the NC2/NC3 rules and for the optimised filters. The stencil size for each value of γ and δ was chosen according to the rules in section 4 and was the same for each rule. The error is calculated at $N = 0$ (forward filter) and at the extreme value $N = 500$ in order to highlight any amplification issues due to successive iterations. The results are shown in Figs. 6 (a) and (b) for the Gaussian and Helmholtz filters respectively.

Simpson's rule performs relatively well for the forward Gaussian/Helmholtz filters, but reconstruction for both filters is unstable. The simple trapezium rule performs generally well for both filtering/reconstruction, and appears to be more robust to variations in the filters' parameters. The optimised filters perform overall better than the trapezium rule both at the filtering and reconstruction levels.

4.5. Optimised inverse filters

From the analysis in section 3 an approximation of the inverse discrete filter transfer function, $\hat{V}_d^N$, is given by,

$$\hat{V}_d^N = \frac{\hat{Q}_d^N}{\hat{G}_d} = \frac{1 - \left(1 - \hat{G}_d\right)^{N+1}}{\hat{G}_d} \tag{17}$$

The inverse is also subject to the consistency condition $\hat{V}_d^N(0) = 1$ while from the above expression we find that $lim_{\hat{G}_d \to 0} \hat{V}_d^N = (N+1)$. If a sufficiently accurate discrete approximation is obtained for $\hat{V}_d^N$, reconstruction may be possible without the need for iteration. This would substantially reduce the computational time for reconstruction making deconvolution-modelling more tractable in LES. In the same spirit of the previous section, we define an optimisation problem to obtain direct-inverse discrete filter coefficients. The optimisation problem reads,

$$\arg\min_{\beta_l} \left(\int_{kh=0}^{\pi} \left(\hat{V}_d^N(kh;\beta_l)\hat{G}_d(kh;g_l) - \hat{Q}_d^N(kh;g_l) \right)^2 d(kh) \right) \tag{18}$$

subject to,

$$\beta_0 + 2\sum_{l=1}^{M_{IF}} \beta_l = 1$$

$$\beta_0 + 2\sum_{l=1}^{M_{IF}} \beta_l cos(kh \cdot l) < (N+1) \ \forall \ kh \in (0, \pi]$$

Note that we have chosen to define the error with respect to $\hat{Q}_d^N$ rather than $\hat{Q}^N$. This ensures that filtering and reconstruction are consistent in the discrete domain regardless of any errors contained in the discrete approximation of the filter. The rationale is that to obtain an accurate direct-inverse filter we must first obtain an accurate enough discrete forward filter. The stencil for the inverse filter, M_{IF}, need not necessarily be equal to the stencil of the forward filter M_{FF}. However, since the aim is to reduce the computational cost by avoiding van Cittert iterations, some analysis is required to derive bounds on the maximum possible size of M_{IF}. For a symmetric forward filter on a $(2M_{FF}+1)$-point stencil, the number of Floating Point Operations (FLOP) (additions, multiplications) required for van Cittert iterations is given by $OP_{FF} = 2N\left(1+(2M_{FF}+1)^D\right)$ where D is the number of dimensions. For a direct-inverse filter the total number of FLOP is given by $OP_{IF} = 2\left(2M_{IF}+1\right)^D - 1$ since no iterations are required for reconstruction. Therefore, provided,

$$\frac{2\left(2M_{IF}+1\right)^D - 1}{2N\left(1+(2M_{FF}+1)^D\right)} < 1 \tag{19}$$

the direct inverse filter will be computationally more efficient than using a forward filter with van Cittert iterations. Therefore there is a limit on the maximum possible M_{IF}. This is proportional to N, M_{FF}, and inversely proportional to D. Typically $N \simeq 5$ at least, and for this value the corresponding $max(M_{IF})$ were calculated and are given in Table 6.

From the results in Table 6 we observe that even for as few as 5 iterations the maximum possible stencil size of the inverse filter can be larger than the stencil of the forward filter before the direct-inverse filter approach becomes computationally more expensive. Even in the 3D case we observe that for $M_{FF} > 4$ M_{IF} can be almost twice the size of M_{FF}. It is also important to note that for the table conditions

D/M_{FF}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	9	14	19	24	29	34	39	44	49	54	59	64	69	74	79	84
2	3	5	7	9	11	14	16	18	20	23	25	27	29	31	34	36
3	2	3	5	7	8	10	12	14	15	17	19	20	22	24	26	27

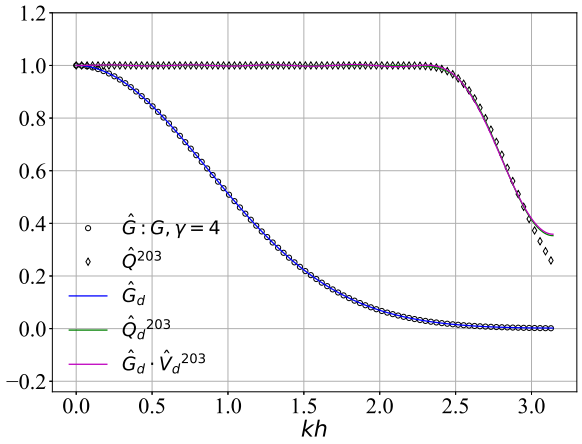
Table 6: Maximum inverse filter half-stencil size M_{IF} obtained using Eq. 19 for $N = 5$ as a function of M_{FF}, and problem dimension D. Below this limit, using a direct-inverse filter (no van Cittert iterations) is computationally less expensive than using a forward filter with $N = 5$ van Cittert iterations for reconstruction.

$(N = 5)$ an accurate enough representation of $\hat{Q}^5$ may be obtained on the same stencil as the for the forward filter which implies that the computational savings using the direct-inverse filter approach can be substantial. In the sections which follow, we show that accurate inverse optimised filters can be obtained on stencil sizes much shorter than the corresponding maximum limits (for the given N in the test-cases) which leads to substantial computational savings.

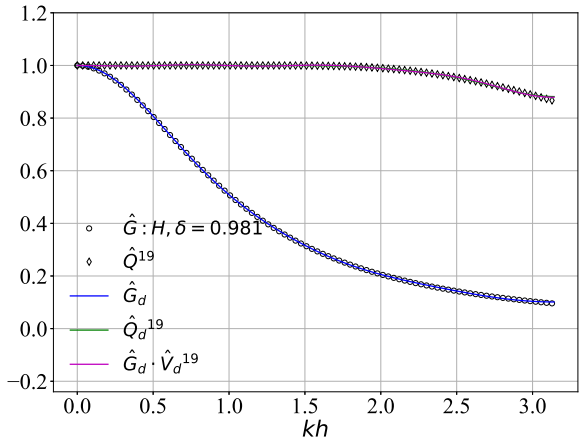
Filter	γ, δ, a	$k_f h$	$(2M_{FF}+1)$	N	$(2M_{IF}+1)$	$e_{opt,FF}$	$e_{opt,IF}$
Gaussian	4	1.017	9	203	21	/	10^{-5}
Helmholtz	0.9807	1.017	13	19	13	/	10^{-5}
Implicit	-0.2618	1.017	11	93	21	10^{-6}	10^{-5}

Table 7: Test-case conditions for obtaining direct-inverse discrete filters: M_{FF} is the forward filter half-stencil size, M_{IF} is the inverse filter half-stencil size, and $e_{opt,FF}$, $e_{opt,IF}$ are the target optimisation errors for the forward and inverse filters respectively.

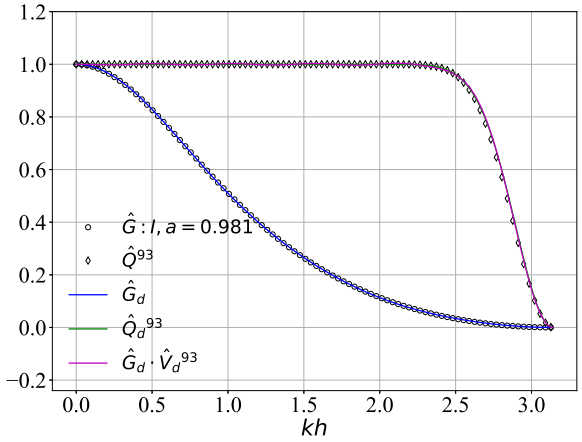
In order to illustrate the method, the optimisation problem was solved and optimised direct-inverse filters obtained. Table 7 lists the conditions for each filter. The filters' parameters were chosen such that all filters have the same k_f. N for each



(a)



(b)



(c)

Fig. 7: Actual filter transfer fucntion ($\hat{G}$) and actual reconstructed transfer function ($\hat{Q}^N$-obtained with van Cittert iterations), discrete transfer function ($\hat{G}_d$) and discrete reconstructed transfer function ($\hat{Q}_d^N$-obtained with van Cittert iterations), and direct-inverse filter transfer function ($\hat{G}_d\hat{V}_d^N$-without van Cittert iterations): (a) for a Gaussian filter with $\gamma = 4$, (b) Helmholtz filter with $\delta = 0.981$, and (c) Implicit filter with $a = -0.262$. The corresponding stencil sizes are given in Table 7.

filter was chosen such that wavenumbers in the range $[0, 0.8\pi]$ are recovered-under these conditions $\hat{Q}_d^N(0.8\pi) > 0.95$ i.e. 80% of the original signal is recovered. For the optimised inverse filters, the adaptive version of the optimisation algorithm was employed with a target error set to 10^{-5}. This choice was found to give good results while keeping the stencil size to a reasonable size-all relevant conditions are given in Table 7. Overall, the Gaussian filter requires the most iterations, and the Helmholtz filter the least since the Helmholtz filter damps larger wavenumbers less hence convergence is faster. The Helmholtz filter also requires the least stencil size for the forward filter making it overall the most efficient to use for reconstruction. The stencil size for the inverse filter is larger for the Gaussian and Implicit filters, and remains the same for the Helmholtz filter. For all three filters, the inverse stencil sizes are much lower than the upper bound predicted by Eq. 19 which implies these are computationally more efficient-in the sections which follow this is quantified and verified for all three filters.

For practical LES applications fewer iterations will be used-the choices here serve as a stringent test for the method. Figure 7 shows the actual transfer functions, the discrete transfer functions, the transfer functions of the reconstructed signal using van Cittert iterations, and the inverse filter transfer functions (without van Cittert iterations) for each filter. Note that $0.8\pi = 2.51$, and all wavenumbers up to this point are recovered by the direct-inverse filters which justifies the use of the proposed optimisation framework. The optimisation framework can be used to derive direct-inverse reconstruction filters for different filter transfer functions and/or iteration counts N depending on the context where reconstruction is to be employed.

4.6. Extension to higher dimensions/arbitrary meshes

One approach for 3D-filtering is to apply 1D filters recursively in each coordinate in which case the filtering operation can be written as,

$$\bar{u}_{i,j,k} = \sum_{n=-M_z}^{M_z} a_n \sum_{m=-M_y}^{M_y} b_m \sum_{l=-M_x}^{M_x} c_l u_{i+l,j+m,k+n}$$

where M_x, M_y, M_z are the half-stencil sizes in each coordinate. Another approach is to define a multi-dimensional filter. These two methods do not necessarily lead to the same multi-dimensional transfer functions. The first method is the most straightforward to implement, and has been applied in numerous studies. Details for the second method are given in [41]. For arbitrary meshes the problem is more difficult since the filter coefficients will depend on the mesh spacing values-a procedure for obtaining such filters is given in [41]. Another approach is to interpolate, filter, and reconstruct on a structured mesh, and then interpolate back on the arbitrary mesh. This, may be computationally more expensive depending on the mesh/geometry. With regards to boundaries, careful attention is required. Even though it is not the focus of this study, skewed filters can be defined near boundaries, and a modified optimisation problem solved. In such cases the transfer function will contain an imaginary component which must be kept small to minimise dispersion errors-for a detailed discussion the reader is referred to [10].

5. 1D test case: model function

The domain for the test-case spans $[0, 1]$, $x_i = ih$, $i \in [0, N_x - 1]$, and $h = 1.0/(N_x - 1)$. The model function is periodic in x,

$$y_i = y(ih) = \sum_{r=0}^{P_x} a_r cos\left(\frac{2\pi r}{N_x h} ih\right) + b_r sin\left(\frac{2\pi r}{N_x h} ih\right)$$

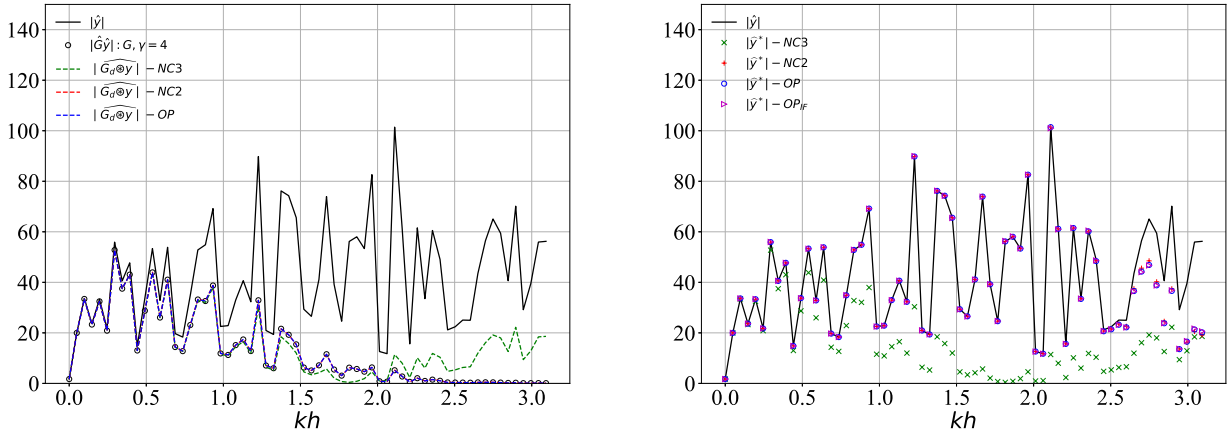
where $P_x = int(N_x/2)$ is the number of modes which include all wavenumbers up to π/h. For this test-case $N_x = 128$, and a_r, b_r are random numbers in the range $[-1, 1]$. The model function is filtered using the Gaussian, Helmholtz, and Implicit filters. Reconstruction is conducted with van Cittert iterations for the NC2, NC3, and optimised forward filters. Reconstruction is also conducted using an optimised inverse filter (without iterations). The conditions for each filter are given in Table 7. These were chosen so as to recover a significant wavenumber range of the original signal as discussed in section 4.5. We note that for the forward filters where van Cittert iterations are employed, M_{FF} is always the same which for the different rules (NC2/NC3/optimised) which ensures a fair comparison between them. To quantify the accuracy of the filtering and reconstruction, the Fourier-mode differences e_f and e_r, are calculated in Fourier space using,

$$e_f = \frac{1}{P_x + 1} \sum_{r=0}^{P_x} \left(|\hat{G}(k_r)\hat{y}(k_r) - \widehat{G_d \circledast y}(k_r)| \right)^2$$

$$e_r = \frac{1}{P_x + 1} \sum_{r=0}^{P_x} \left(|\hat{y}(k_r)| - |\widehat{y^*}(k_r)| \right)^2$$

where $\hat{}$ denotes the discrete Fourier transform. Figure 8 (a) shows the original and Gaussian-filtered test function in Fourier-space using the different rules, and Fig. 8 (b) shows the corresponding Fourier transforms of the reconstructed fields. Figures 9 and 10 show the corresponding results for the Helmholtz-filtered and Implicit-filtered functions respectively. The errors e_f, e_r are given in Table 8. Table 9 additionally shows the corresponding FLOP ratios for each filter obtained using Eq. 19, and the actual computational time ratios required for reconstruction: T_{FF} is the computational time for a forward filter (with van Cittert iterations), and T_{IF} is the computational time for a direct-inverse optimised filter (without iterations). Note

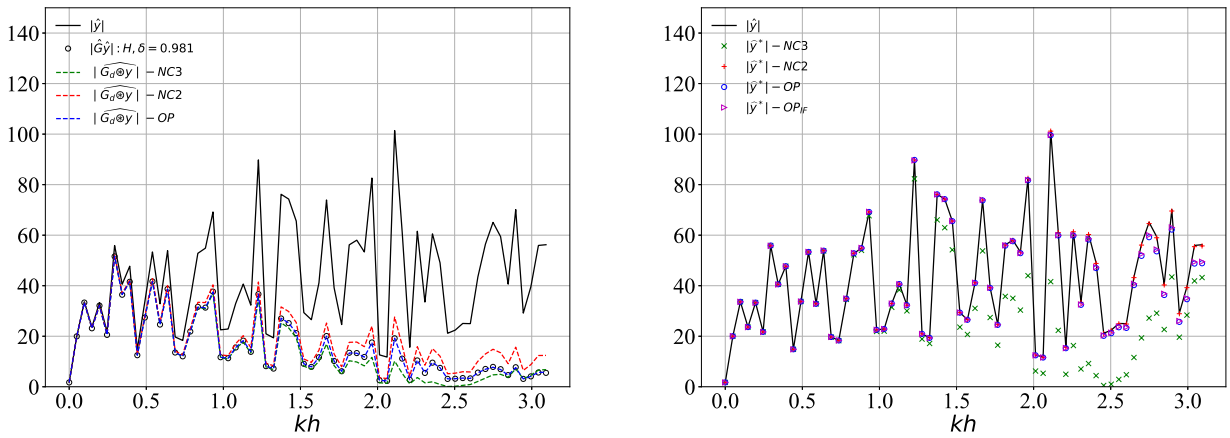
that the computational times for the forward filters are about the same for the NC3/NC2/optimised filters since they all have the same stencil. In order to increase the statistical accuracy of the results, the computational timings were averaged over 1000 reconstructions, and all compiler optimisations were switched off.



(a) (b)

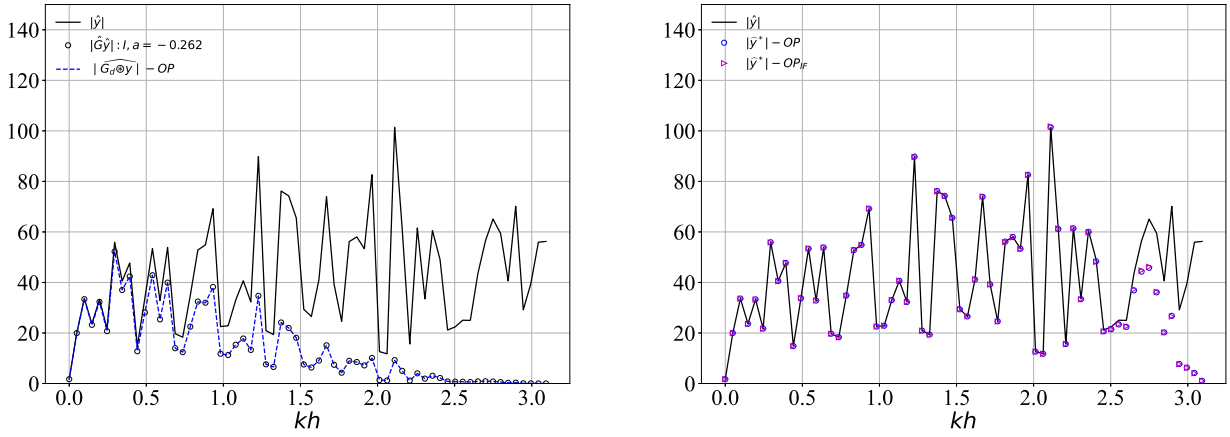
Fig. 8: (a) Absolute value of the discrete Fourier transform of the test-function, and of the Gaussian-filtered ($\gamma = 4$) test-function using the different rules, NC3 (Simpson's rule), NC2 (trapezium rule), OP (Optimised filter), and (b) corresponding transforms of the reconstructed functions for each rule including the optimised direct-inverse filter ($\mathrm{OP_{IF}}$) with no van Cittert iterations. The corresponding filter parameters for each filter are given in Table 7. Filtering damps the high-wavenumber components of the test-function as seen in (a), and reconstruction recovers a significant range of these wavenumbers as seen in (b) thus recovering to a good extent the original function.

For the Gaussian filter, the optimised filters perform well both for filtering and reconstruction. For filtering, the error is lower than for the NC2 rule and for reconstruction the errors are similar. The optimised inverse filter also performs well, however at a much reduced computational cost-about 100 times lower which is in



(a) (b)

Fig. 9: As in Fig. 8 for the Helmholtz-filtered ($\delta = 0.981$) test-function.



(a) (b)

Fig. 10: As in Fig. 8 for the Implicit-filtered ($\alpha = -0.262$) function.

e_f

Filter	NC3	NC2	OP
Gaussian	$1.412{\cdot}10^{1}$	$2.692{\cdot}10^{-4}$	$1.829{\cdot}10^{-5}$
Helmholtz	1.806	3.461	$1.458{\cdot}10^{-3}$
Implicit	/	/	$3.210{\cdot}10^{-4}$

e_r

Filter	NC3	NC2	OP	OP-IF
Gaussian	$5.990{\cdot}10^{2}$ (Unstable)	$5.469{\cdot}10^{1}$	$5.479{\cdot}10^{1}$	$5.406{\cdot}10^{1}$
Helmholtz	$1.967{\cdot}10^{2}$ (Unstable)	$1.848{\cdot}10^{-2}$	3.193	2.551
Implicit	/	/	$1.090{\cdot}10^{2}$	$1.091{\cdot}10^{2}$

Table 8: Filtering and reconstruction errors for the 1D test-case.

good agreement with the actual FLOP count ratio predicted by Eq. 19. The NC3 rule performs poorly both for filtering/reconstruction in accordance with the results in the previous sections: $\hat{G}_d < 0$ for this rule which leads to unstable reconstruction causing the reconstruction algorithm to exit prematurely.

For the Helmholtz filter, the NC3 rule performs better than the NC2 rule for filtering, while the optimised filters perform better in comparison. For reconstruction, the error for the NC3 rule is quite large for the same reasons as above. The NC2 rule appears to have the lowest reconstruction error, however this is misleading. Fig. 9(a) shows that the NC2 rule is in fact a poor approximation of the actual filter transfer function. This is why the filtering error for the NC2 rule shown in Table 8 is larger than for the rest of the rules. In fact, the NC2 rule approximates $\hat{G}$ so poorly that damping is much lower than what should be for all wavenumbers on the mesh. As a result, for the same number of iterations, the reconstruction error is lower simply because of the lower damping at the filtering level. This is also evident in

Fig. 9(b) where the NC2 rule actually recovers almost all wavenumbers on the mesh something which it was not supposed to do. The optimised filter on the other hand is a good approximation of $\hat{G}$, and performs well for reconstruction. The inverse filter also performs well, and is more efficient as evident by the results in Table 9- about 20 times faster. Similarly good results are obtained for the Implicit filter. A slight difference in this case is that in accordance with the optimisation conditions for this filter, the grid cut-off is completely damped, and remains damped during reconstruction.

Filter	γ, δ, a	M_{FF}	M_{IF}	N	OP_{FF}/OP_{IF}	Time ratio T_{FF}/T_{IF}
Gaussian	4	4	10	203	99.02	109.23
Helmholtz	0.9807	6	6	19	21.28	20.40
Implicit	-0.2618	5	10	93	54.44	60.49

Table 9: Predicted and measured computational time ratios for the 1D test-case using a forward filter (FF) with van Cittert iterations, and the optimised inverse filter (IF) without any iterations.

6. 3D test case: turbulent flames

The data correspond to a 3D Direct Numerical Simulation (DNS) database of a turbulent premixed flame propagating in an inflow-outflow configuration. Cold reactants enter from one end of the computational domain and hot products leave from the other end. The DNS was conducted using the SENGA2 code [42] which solves the reacting Navier-Stokes equations in fully compressible form, using a 10th order finite-difference scheme for the interior spatial derivatives, and a 4th-order explicit Runge-Kutta scheme for the time-stepping. The particular database simulates the turbulent combustion of a multi-component syngas-like fuel. Chemistry is modelled

using a detailed chemical mechanism specifically developed for complex fuels [43, 44]. Details of the simulation geometry/parameters, and the DNS solver, can be found in [42, 45].

Mesh	L_x x L_y x L_z (mm)	N_x x N_y x N_z	γ
h_o-DNS	14.0 x 7.0 x 7.0	768 x 384 x 384	/
h_1-LES	Same	97 x 49 x 49	4.0

Table 10: DNS and LES meshes used for the 3D test-case.

In order to simulate an LES, the DNS data are sampled on a coarser mesh h_1 using Lagrange interpolation. On the LES mesh, the fields of interest (obtained by interpolating from the DNS on the LES mesh) are the reference fields. These reference fields are filtered on the LES mesh, and are then reconstructed on the LES mesh. This procedure is different than the usual approach of filtering on the DNS mesh and interpolating on the LES mesh because the LES mesh resolves adequately enough the fields of interest. In this approach we avoid the expensive procedure of filtering every time on the DNS mesh.

The primary field of interest is density which is used to obtain primitive variables from conservative ones [28]. It is important to note that in the current implementation of the direct-inverse filter approach, there is no explicit bounds' control on the reconstructed signal whereas bounds' control can be implemented when using van Cittert iterations through a dynamic b [28]. Therefore, in order to examine how well direct-inverse filter reconstruction performs for scalars we additionally consider the mass fraction of O_2, a major species. Mass fractions are reconstructed using $y_k^* = \{\rho y_k\}^*/\rho^*$ following previous works [28]. When using van Cittert iterations, the classical approach is considered with no bounds control to ensure a fair comparison is made.

Details of the DNS and LES meshes are given in Table 10. All filters have the same k_f. In particular, $\gamma_{h_1} = \Delta/h_1 = 4$, $\delta_{h_1} = \lambda/h_1 = 0.981$, and $a = -0.262$ as shown in Table 2, and they damp substantially LES-resolved wavenumbers. 3D filtering on the LES mesh is conducted using dimensional splitting as explained in section 4.6. Reconstruction is conducted using the different rules excluding however Simpson's rule for which reconstruction was unstable as shown in the 1D test-case. In contrast to the 1D test-case, reconstruction is conducted to recover wavenumbers only in the range $[0, 2k_f]$ which is more than enough to recover k_f i.e. as would be done in actual LES. The corresponding N required for each filter are given in Table 2. Optimised inverse filters (without van Cittert iterations) at the corresponding N for each filter given in Table 2 were also computed, and the filtered fields reconstructed using this approach as well.

Filter	NC3	NC2	OP	OP-IF
Gaussian	Unstable	$1.742{\cdot}10^{-6}$	$1.762{\cdot}10^{-6}$	$3.566{\cdot}10^{-6}$
Helmholtz	Unstable	$5.251{\cdot}10^{-7}$	$1.388{\cdot}10^{-6}$	$6.801{\cdot}10^{-8}$
Implicit	/	/	$1.734{\cdot}10^{-6}$	$8.728{\cdot}10^{-7}$

Table 11: Spatial reconstruction errors e_ρ for the different rules.

Filter	γ, δ, a	M_{FF}	M_{IF}	N	OP_{FF}/OP_{IF}	Time ratio T_{FF}/T_{IF}
Gaussian	4	4	7	46	9.95	10.08
Helmholtz	0.9807	6	6	13	13.01	12.86
Implicit	-0.2618	5	5	26	26.03	25.96

Table 12: Predicted and measured computational time ratios for the 3D test-case using a forward filter (FF) with van Cittert iterations, and the optimised inverse filter (IF) without any iterations.

The reconstruction error is calculated using,

$$e_u = \frac{1}{N_x N_y N_z} \sum_{k=1}^{N_z} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} ({u^*}_{i,j,k} - u_{i,j,k})^2$$

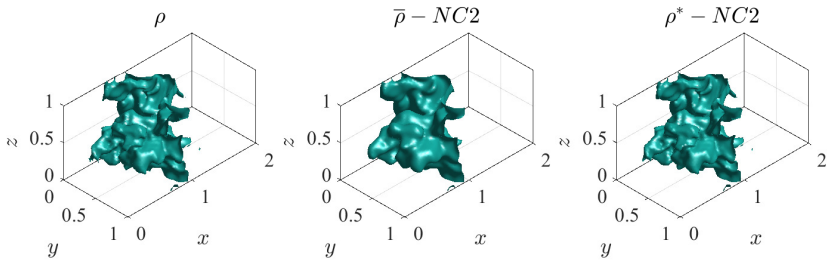
where N_x, N_y, and N_z are the number of points for the LES mesh.

Figures 11(a)-(c) show iso-surfaces of the the original, Gaussian-filtered and reconstructed density using the NC2 rule, optimised filter, and optimised inverse filter respectively on the LES mesh. Similar results were obtained using the Helmholtz and Implicit filters. Note that the filtered field in Figs. 11(b) and (c) is the same because in both cases the optimised forward filter was used. As expected, filtering smooths the density field by removing high-wavenumber components. Reconstruction on the other hand recovers some of the damped wavenumbers on the LES mesh thereby "sharpening" the density field. Figure 12 additionally shows for each mesh-point of the LES mesh the reconstructed density against the reference density using the different approaches-note that these values are normalised using $\rho^+ = (\rho - \rho_{min})/(\rho_{max} - \rho_{min})$. The Pearson correlation coefficients are near-unity for all three cases, and the agreement is good for the entire range of reference density values. It is also important to note that for this particular database there are strong density gradients because of combustion. These gradients occur over a small region in space where heat is being released [45]. In fact, ρ^+ indicates the extent of reaction with $\rho^+ = 1$ corresponding to unburnt cold gases, and $\rho^+ = 0$ corresponding to hot burnt gases. In between these values reaction takes place, and large density gradients exist. The scatter plots indicate that even in regions with large gradients e.g. around $\rho^+ = 0.5$, the reconstructed density is well captured both with the optimised filter, and with the optimised inverse filter. Table 11 shows the corresponding spatial reconstruction errors for each filter/rule. For the Gaussian filter, the reconstruction errors are similar for all three rules. For the Implicit filter, the optimised forward and inverse

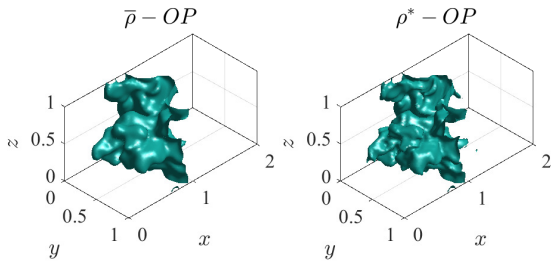
filters perform well. Again the NC2 rule appears to provide improved results due to the lower damping caused by the poor approximation of the Helmholtz filter as in the 1D test-case. It is interesting to note that the optimised inverse filter for the Helmholtz and Implicit filters improves reconstruction.

Figure 13 shows the corresponding reconstructed oxygen mass fraction for the Gaussian filter and using the different rules. The red squares indicate the minimum and maximum values of the reference mass fractions. The scatter plots indicate that for all three reconstruction rules the reconstructed fields are in good agreement with the reference fields with no significant non-physical under/over predictions even though there is no explicit bounds' control even when using the direct-inverse filter approach. Similar results were obtained for the Helmholtz and Implicit filters.

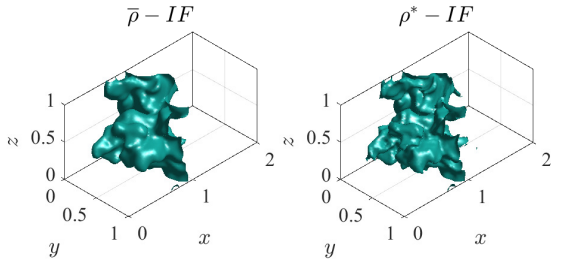
In terms of computational efficiency, Table 12 shows the corresponding FLOP ratios and actual measured computational time ratios between the forward filter approach with van Cittert iterations, and the inverse filter approach without any iterations. The measured computational time ratios for the 3D test-case are in good agreement with the FLOP ratios as predicted by Eq. 19. Even though the gain is inversely proportional to the number of dimensions, we observe that the computational time savings for the 3D test-case by using the inverse filter approach are still substantial.



(a)

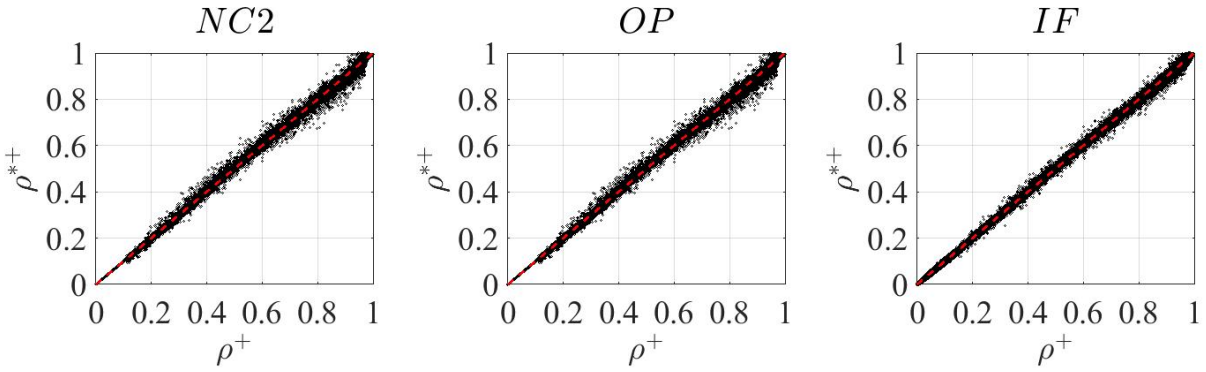


(b)



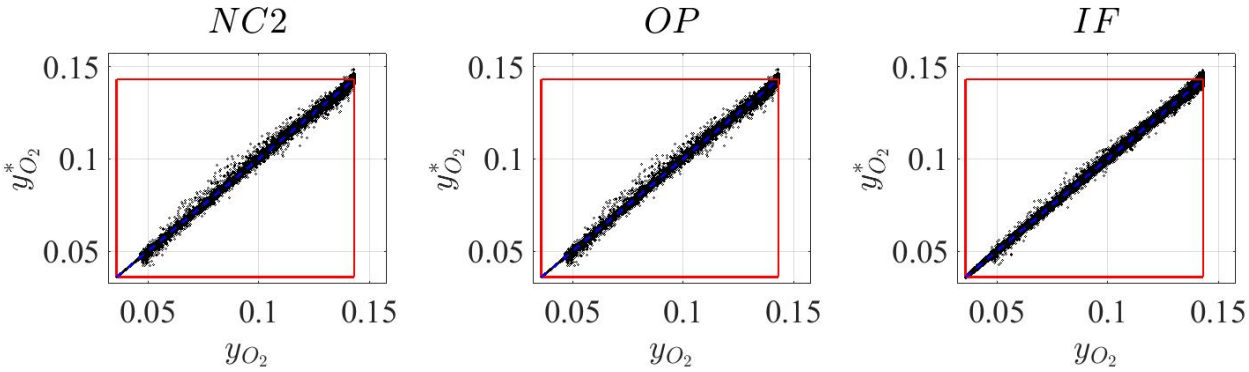
(c)

Fig. 11: Density iso-surface on the LES mesh (value=0.4) of original reference density (ρ), Gaussian-filtered density at $\gamma = 4$ ($\bar{\rho}$), and reconstructed density (ρ^*), using the different rules: NC2 (trapezium rule), OP (Optimised filter), IF (inverse filter without van Cittert iterations). Filtering smooths the original reference field, and reconstruction recovers the original field thereby ρ^* resembles ρ on the LES mesh.



(a) (b) (c)

Fig. 12: Scatter plots of normalised reconstructed density (filter=Gaussian, $\gamma = 4$) against normalised reference density on the LES mesh using: (a) NC2 rule with van Cittert iterations, (b) using the optimised filter with van Cittert iterations, and (c) using the optimised inverse filter without van Cittert iterations.



(a) (b) (c)

Fig. 13: Scatter plots of reconstructed y_{O_2} (mass fraction) (filter=Gaussian, $\gamma = 4$) against reference y_{O_2} on the LES mesh using: (a) NC2 rule with van Cittert iterations, (b) using the optimised filter with van Cittert iterations, and (c) using the optimised inverse filter without van Cittert iterations.

7. Conclusions

The reconstruction properties of the explicit Gaussian, Helmholtz, and of the 3-point Implicit (Padé) filter using the van Cittert deconvolution algorithm are investigated analytically. The von Neumann analysis leads to a necessary stability condition for the discrete filter transfer function $\hat{G}_d$. The different rules/methods considered for obtaining discrete filters include: (a) Newton-Cotes formulas, (b) filters based on Taylor-expansions, and (c) filters derived by solving a newly-proposed constrained and adaptive optimisation problem. The main results are as follows,

(i) Higher-order Newton-Cotes rules on a given stencil improve the accuracy of $\hat{G}_d$ at low wavenumbers but reduce the accuracy at higher wavenumbers to the extent that $\hat{G}_d$ may become negative leading to unstable reconstruction.

(ii) The simple trapezium rule was found to be robust with stable reconstruction for the Gaussian/Helmholtz filters, and for all filter parameter values considered in this study.

(iii) Simpson's rule improves the results at the filtering level for the Helmholtz filter but not for the Gaussian filter. At the reconstruction level, Simpson's rule may result in $\hat{G}_d < 0$ for some wavenumbers, depending on the filter parameter value, which leads to unstable reconstruction.

(iv) The proposed optimisation framework in section 4.4 produces forward filters which are consistent and stable during reconstruction, and which offer improved accuracy at the filtering and reconstruction levels.

(iv) Reconstruction using the optimised direct-inverse filters using the optimisation framework in section 4.5 is stable, accurate, and removes the need to apply successive van Cittert/otherwise iterations which substantially reduces the computational cost for reconstruction.

Discrete optimised forward and inverse coefficients for the Gaussian, Helmholtz, and 3-point-Implicit filters for different values of their corresponding parameters are given as supplementary material. Even though the context of this work was for modelling purposes in LES, these filters may be used in a variety of digital signal-processing applications. In addition, the proposed optimisation frameworks are flexible, and the optimisation parameters/targets/constraints are straightforward to modify to suit the users' needs. The relevant optimisation routines can be made available by contacting the authors directly.

Acknowledgement

The REDAFLOW project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101019855.

References

[1] P. Sagaut, Large Eddy Simulation for Incompressible Flows: An Introduction, Springer-Verlag (2001).

[2] S. Lele, Compact finite-difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[3] H. Pitsch, Large Eddy Simulation of Turbulent Combustion, Ann. Rev. Fluid Mech. 38 (2006) 453–482.

[4] L. Y. M. Gicquel, G. Staffelbach, T. Poinsot, Large Eddy Simulations of gaseous flames in gas turbine combustion chambers, Prog. Energy Combust. Sc. 38 (2012) 782–817.

[5] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.

[6] B. Geurts, Inverse modelling for large-eddy simulation, Phys. Fluids 9 (1997) 3585–3587.

[7] J. G. M. Kuerten, B. Geurts, A. Vreman, M. Germano, Dynamic inverse modelling and its testing in large-eddy simulations of the mixing layer, Phys. Fluids 11 (1999) 3778–3785.

[8] J. Domaradzki, E. Saiki, A subgrid-scale model based on the estimation of unresolved scales of turbulence, Phys. Fluids 2148-2164 (1997).

[9] S. Stolz, N. Adams, An approximate deconvolution procedure for large-eddy simulation, Phys. Fluids 11 (1999) 1699–1701.

[10] S. Stolz, N. Adams, An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows, Phys. Fluids 13 (2001) 997–1015.

[11] N. Adams, S. Stolz, A sub-grid scale deconvolution approach for shock-capturing, J. Comput. Phys. 178 (2002) 391–426.

[12] P. V. Cittert, Zum Einfluss der Spaltbreite auf die Intensitatsverteilung in Spektrallinien. II, Zeitschrift fur Physik 69 (1931) 298–308.

[13] O. A. Mahfoze, S. Laizet, Non-explicit large eddy simulations of turbulent channel flows from Re_τ=180 up to Re_τ= 5200, Comput. Fluids 228 (2021) 105019.

[14] A. Aprovitola, F. Denaro, On the application of congruent upwind discretizations for large eddy simulations, J. Comput. Phys. 194 (2004) 329–343.

[15] N. Adams, S. Hickel, S. Franz, Implicit subgrid-scale modeling by adaptive deconvolution, J. Comput. Phys. 200 (2004) 412–431.

[16] S. Hickel, N. Adams, J. Domaradzki, An adaptive local deconvolution method for implicit LES, J. Comput. Phys. 213 (2006) 413–436.

[17] F. Schwertfirm, J. Mathew, M. Manhart, Improving spatial resolution characteristics of finite difference and finite volume schemes by approximate deconvolution pre-processing, Comput. Fluids 37 (2008) 1092–1102.

[18] A. Boguslawski, K. Wawrzak, A. Paluszewska, B. J. Geurts, Deconvolution of induced spatial discretization filters subgrid modeling in LES: application to two-dimensional turbulence, J. Phys. 2090 (2021) 1–10.

[19] C. D. Pruett, A temporally regularized buffer domain for flow-through simulations, Comput. Fluids 109 (2015) 35–48.

[20] P. Nathen, M. Haussmann, M. Krause, N. Adams, Adaptive filtering for the simulation of turbulent flows with lattice Boltzmann methods, Comput. Fluids 172 (2018) 510–523.

[21] J. Mathew, Large eddy simulation of a premixed flame with approximate deconvolution modelling, Proc. Combust. Inst. 29 (2002) 1995–2000.

[22] P. Domingo, L. Vervisch, Large Eddy Simulation of premixed turbulent combustion using approximate deconvolution and explicit flame filtering, Proc. Combust. Inst. 35 (2015) 1349–1357.

[23] P. Domingo, L. Vervisch, DNS and approximate deconvolution as a tool to analyse one-dimensional filtered flame sub-grid scale modelling, Combust. Flame 177 (2017) 109–122.

[24] C. Mehl, B. Fiorina, Evaluation of deconvolution modelling applied to numerical combustion, Combust. Th. Model. 22 (2017) 38–70.

[25] Q. Wang, M. Ihme, Regularized deconvolution method for turbulent combustion modelling, Combust. Flame 176 (2017) 125–142.

[26] Q. Wang, M. Ihme, A regularised deconvolution method for turbulent closure modelling in implicitly filtered large-eddy simulation, Combust. Flame 204 (2019) 341–355.

[27] Z. M. Nikolaou, L. Vervisch, R. S. Cant, Scalar flux modelling in turbulent flames using iterative deconvolution, Phys. Rev. Fluids 3 (2018) 043201.

[28] Z. M. Nikolaou, L. Vervisch, A priori assessment of an iterative deconvolution method for LES sub-grid scale variance modelling, Flow Turb. Combust. 101 (2018) 33–53.

[29] A. Seltz, P. Domingo, L. Vervisch, Z. Nikolaou, Direct mapping from LES resolved scales to filtered-flame generated manifolds using convolutional neural networks, Combust. Flame 210 (2019) 71–82.

[30] P. Domingo, Z. Nikolaou, A. Seltz, L. Vervisch, From Discrete and Iterative Deconvolution Operators to Machine Learning for Premixed Turbulent Combustion Modeling, Springer International Publishing, Cham, 2020, pp. 215–232. doi:10.1007/978-3-030-44718-2_11.
URL https://doi.org/10.1007/978-3-030-44718-2_11

[31] J. Bull, A. Jameson, Explicit filtering and exact reconstruction of the sub-filter stresses in large eddy simulation, J. Comput. Phys. 306 (2016) 117–136.

[32] O. San, A. Staples, T. Iliescu, A posteriori analysis of low-pass spatial filters for approximate deconvolution large eddy simulations of homogeneous incompressible flows, Int. J. Comput. Fluid Dyn. 29 (2015) 40–66.

[33] O. San, Analysis of low-pass filters for approximate deconvolution closure modelling in one-dimensional decaying Burgers turbulence, Int. J. Comput. Fluid Dyn. 30 (2016) 20–37.

[34] A. Povitsky, P. Morris, A Higher-Order Compact Method in Space and Time Based on Parallel Implementation of the Thomas Algorithm, J. Comput. Phys. 161 (2000) 182–203.

[35] P. Schlatter, S. Stolz, L. Kleiser, LES of transitional flows using the approximate deconvolution model, Int. J. Heat and Fluid Flow 25 (2004) 549–558.

[36] O. San, P. Veluda, Generalized deconvolution procedure for structural modelling of turbulence, J. Sci. Comput. 75 (2018) 1187–1206.

[37] W. Layton, M. Neda, A similarity theory of approximate deconvolution models of turbulence, J. Math. Anal. Appl. 333 (2007) 416–429.

[38] W. Layton, I. Stanculescu, Chebyshev optimised approximate deconvolution models of turbulence, Appl. Math. and Comput. 208 (2009) 106–118.

[39] L. Berselli, R. Lewandowski, Convergence of approximate deconvolution models to the mean Navier-Stokes equations, Ann. I.H. Poincare 29 (2012) 171–198.

[40] A. Dunca, Estimates of the discrete van Cittert deconvolution error in approximate deconvolution models of turbulence in bounded domains, Appl. Num. Math 134 (2018) 1–10.

[41] P. Sagaut, R. Grohens, Discrete filters for large eddy simulation, Int. J. Num. Meth. Fluids 31 (1999) 1195–1220.

[42] R. S. Cant, SENGA2 User Guide, CUED A-THERMO-TR67 (2012).

[43] Z. M. Nikolaou, N. Swaminathan, A 5-step reduced mechanism for combustion of $CO/H_2/H_2O/CH_4/CO_2$ mixtures with low hydrogen/methane and high H_2O content, Combust. Flame 160 (2013) 56–75.

[44] Z. M. Nikolaou, N. Swaminathan, Evaluation of a reduced mechanism for turbulent premixed combustion, Combust. Flame 161 (2014) 3085–3099.

[45] Z. M. Nikolaou, N. Swaminathan, Direct numerical simulation of complex fuel combustion with detailed chemistry: physical insight and mean reaction rate modelling, Combust. Sc. Tech. 187 (2015) 1759–1789.

Appendix

8. Discrete filters based on Taylor expansion

Gaussian	G3	G5	G7	G9
g_0	$\frac{\left(12-\gamma^2\right)}{12}$	$\frac{\left(192-20\gamma^2+\gamma^4\right)}{192}$	$\frac{\left(20736-2352\gamma^2+168\gamma^4-5\gamma^6\right)}{20736}$	$\frac{\left(3981312-472320\gamma^2+39312\gamma^4-1800\gamma^6+35\gamma^8\right)}{3981312}$
g_1	$\frac{\gamma^2}{24}$	$\frac{\left(16\gamma^2-\gamma^4\right)}{288}$	$\frac{\left(1728\gamma^2-156\gamma^4+5\gamma^6\right)}{27648}$	$\frac{\left(331776\gamma^2-35136\gamma^4+1740\gamma^6-35\gamma^8\right)}{4976640}$
g_2		$\frac{\left(-4\gamma^2+\gamma^4\right)}{1152}$	$\frac{\left(-432\gamma^2+120\gamma^4-5\gamma^6\right)}{69120}$	$\frac{\left(-82944\gamma^2+24336\gamma^4-1560\gamma^6+35\gamma^8\right)}{9953280}$
g_3			$\frac{\left(192\gamma^2-60\gamma^4+5\gamma^6\right)}{414720}$	$\frac{\left(36864\gamma^2-12096\gamma^4+1260\gamma^6-35\gamma^8\right)}{34836480}$
g_4				$\frac{\left(-20736\gamma^2+7056\gamma^4-840\gamma^6+35\gamma^8\right)}{278691840}$

Table 13: Discrete Gaussian-filter coefficients on 3, 5, 7, and 9-point stencils obtained from Taylor expansion ($\gamma = \Delta/h$).

Helmholtz	H3	H5	H7	H9
g_0	$1-2\delta^2$	$\frac{\left(12-30\delta^2+72\delta^4\right)}{12}$	$\frac{\left(180-490\delta^2+1680\delta^4-3600\delta^6\right)}{180}$	$\frac{\left(5040-14350\delta^2+57330\delta^4-189000\delta^6+352800\delta^8\right)}{5040}$
g_1	δ^2	$\frac{\left(16\delta^2-48\delta^4\right)}{12}$	$\frac{\left(270\delta^2-1170\delta^4+2700\delta^6\right)}{180}$	$\frac{\left(8064\delta^2-40992\delta^4+146160\delta^6-282240\delta^8\right)}{5040}$
g_2		$\frac{\left(-\delta^2+12\delta^4\right)}{12}$	$\frac{\left(-27\delta^2+360\delta^4-1080\delta^6\right)}{180}$	$\frac{\left(-1008\delta^2+14196\delta^4-65520\delta^6+141120\delta^8\right)}{5040}$
g_3			$\frac{\left(2\delta^2-30\delta^4+180\delta^6\right)}{180}$	$\frac{\left(128\delta^2-2016\delta^4+15120\delta^6-40320\delta^8\right)}{5040}$
g_4				$\frac{\left(-9\delta^2+147\delta^4-1260\delta^6+5040\delta^8\right)}{5040}$

Table 14: Discrete Helmholtz-filter coefficients on 3, 5, 7, and 9-point stencils obtained from Taylor expansion ($\delta = \lambda/h$).

Box	B3	B5	B7	B9
g_0	$\frac{\left(12-\epsilon^2\right)}{12}$	$\frac{\left(960-100\epsilon^2+3\epsilon^4\right)}{960}$	$\frac{\left(241920-27440\epsilon^2+1176\epsilon^4-15\epsilon^6\right)}{241920}$	$\frac{\left(46448640-5510400\epsilon^2+275184\epsilon^4-5400\epsilon^6+35\epsilon^8\right)}{46448640}$
g_1	$\frac{\epsilon^2}{24}$	$\frac{\left(80\epsilon^2-3\epsilon^4\right)}{1440}$	$\frac{\left(6720\epsilon^2-364\epsilon^4+5\epsilon^6\right)}{107520}$	$\frac{\left(3870720\epsilon^2-245952\epsilon^4+5220\epsilon^6-35\epsilon^8\right)}{58060800}$
g_2		$\frac{\left(-20\epsilon^2+3\epsilon^4\right)}{5760}$	$\frac{\left(-336\epsilon^2+56\epsilon^4-1\epsilon^6\right)}{53760}$	$\frac{\left(-967680\epsilon^2+170352\epsilon^4-4680\epsilon^6+35\epsilon^8\right)}{116121600}$
g_3			$\frac{\left(448\epsilon^2-84\epsilon^4+3\epsilon^6\right)}{967680}$	$\frac{\left(61440\epsilon^2-12096\epsilon^4+540\epsilon^6-5\epsilon^8\right)}{58060800}$
g_4				$\frac{\left(-34560\epsilon^2+7056\epsilon^4-360\epsilon^6+5\epsilon^8\right)}{464486400}$

Table 15: Discrete Box-filter coefficients on 3, 5, 7, and 9-point stencils obtained from Taylor expansion ($\epsilon = L/h$).

9. Optimised forward filter coefficients

Optimised forward filter coefficients (g_l) and direct-inverse filter coefficients (b_l) are given below. The filtering/reconstruction relations are,

$$\bar{u}_i = \sum_{l=-M_{FF}}^{M_{FF}} g_l u_{i+l} \qquad\qquad {u_i}^* = \sum_{l=-M_{IF}}^{M_{IF}} b_l \bar{u}_{i+l}$$

The optimised direct-inverse filters were obtained at $N = 5$ van Cittert iterations. $\gamma = \Delta/h$ for the Gaussian filter, and $\delta = \lambda/h$ for the Helmholtz filter. The filter parameter range was chosen to reflect practical values commonly used in the literature. Note that for each forward filter, and for a given filter parameter, the corresponding inverse filter is given for the said forward filter parameter in the inverse filter table. M_{FF} for the Gaussian/Helmholtz filters was chosen according to the rules in section 4. M_{FF} for the Implicit filter was obtained dynamically by setting a target optimisation error of 10^{-6}. M_{IF} for the direct-inverse filters is always obtained dynamically for all three filters by setting a target error of 10^{-5}. The error listed in the tables corresponds to the final mean-squared error between the optimised discrete forward/reconstructed transfer functions and the actual forward/reconstructed transfer functions. The optimisation settings, target errors, reconstruction iterations N etc. may be modified to suit the user's needs. Note that the damping efficiency η in the Tables is defined as,

$$\eta = 1 - \frac{1}{\pi} \int_{kh=0}^{kh=\pi} \hat{G}(kh) d(kh)$$

and that for the inverse filters the quoted $k_f h, \eta$ are for the reoconstructed transfer function $\hat{V}_d^N \hat{G}_d$ which serve to quantify the extent of reconstruction.

Forward filter:	**Gaussian**				
γ	4	5	6	7	8
$k_f h$	1.016	0.813	0.679	0.580	0.508
η	0.655	0.724	0.770	0.803	0.827
$\hat{G}_d(\pi)$	2.134E-03	3.428E-05	2.027E-04	1.773E-09	2.181E-04
M_{FF}	4	5	6	7	8
Error	2.129E-08	6.196E-09	1.298E-08	2.064E-08	2.593E-08
g0	3.45415481E-01	2.76409458E-01	2.30361794E-01	1.97475161E-01	1.72812355E-01
g1	2.37565592E-01	2.17425584E-01	1.94962542E-01	1.74652668E-01	1.57270243E-01
g2	7.70135187E-02	1.05845990E-01	1.18286620E-01	1.21008168E-01	1.18750472E-01
g3	1.19009368E-02	3.18779064E-02	5.13942706E-02	6.55970380E-02	7.43261643E-02
g4	8.12212165E-04	5.95785198E-03	1.60156429E-02	2.78307565E-02	3.85520948E-02
g5		6.87938911E-04	3.59250436E-03	9.27026705E-03	1.65816852E-02
g6			5.67523443E-04	2.42349568E-03	5.93542954E-03
g7				4.80026714E-04	1.76737453E-03
g8					4.10358526E-04
Inverse filter: N=5					
γ	4	5	6	7	8
$k_f h$	1.825	1.461	1.213	1.043	0.912
η	0.409	0.526	0.605	0.661	0.704
$\hat{V}_d^N \hat{G}_d(\pi)$	1.117E-02	2.057E-04	3.565E-04	8.704E-09	3.786E-04
M_{IF}	4	5	6	7	8
Error	6.519E-06	4.192E-06	1.813E-06	1.121E-06	7.857E-07
b0	3.48544218E+00	3.94893539E+00	3.81406144E+00	3.70253496E+00	3.62257966E+00
b1	-1.24814783E+00	-1.24731324E+00	-7.16575919E-01	-3.94376008E-01	-2.01146597E-01
b2	-1.55292944E-01	-3.75990613E-01	-9.75288675E-01	-9.97560350E-01	-8.59013025E-01
b3	1.89248380E-01	3.67688340E-02	2.95305166E-01	-2.11596765E-01	-5.27024873E-01
b4	-2.85287002E-02	1.51522917E-01	-1.72149686E-01	3.24836028E-01	1.55763485E-01
b5		-3.94555931E-02	2.31596699E-01	-2.72938255E-01	1.91402698E-01
b6			-6.99183036E-02	2.98632698E-01	-2.98558392E-01
b7				-9.82648256E-02	3.52884957E-01
b8					-1.25598080E-01

Forward filter:	**Helmholtz**				
δ	0.9	1.0	1.1	1.2	1.3
$k_f h$	1.110	0.998	0.908	0.831	0.769
η	0.565	0.598	0.627	0.652	0.674
$\hat{G}_d(\pi)$	1.177E-01	9.618E-02	8.054E-02	6.897E-02	5.854E-02
M_{FF}	7	8	8	9	10
Error	1.491E-06	3.860E-07	3.962E-07	4.630E-07	1.312E-07
g0	4.35408387E-01	4.01919378E-01	3.73081451E-01	3.47935525E-01	3.25847440E-01
g1	1.95604785E-01	1.94464743E-01	1.92050952E-01	1.88703449E-01	1.84769063E-01
g2	5.60486232E-02	6.41254101E-02	7.08121613E-02	7.61701017E-02	8.03697299E-02
g3	2.19610597E-02	2.66355134E-02	3.12474636E-02	3.55215537E-02	3.93857877E-02
g4	5.41625050E-03	8.16549348E-03	1.11748563E-02	1.41857177E-02	1.71223739E-02
g5	3.01695761E-03	4.04181145E-03	5.43905280E-03	7.00217254E-03	8.65962773E-03
g6	2.48130493E-04	7.93446697E-04	1.60735893E-03	2.52473892E-03	3.54236379E-03
g7		8.13893018E-04	1.12742972E-03	1.53077020E-03	2.01029212E-03
g8				3.93733297E-04	6.76369324E-04
g9					5.40672483E-04
Inverse filter: N=5					
δ	0.9	1	1.1	1.2	1.3
$k_f h$	N/A	2.840	2.607	2.387	2.198
η	0.155	0.193	0.229	0.263	0.296
$\hat{V}_d^N \hat{G}_d(\pi)$	5.280E-01	4.538E-01	3.944E-01	3.491E-01	3.030E-01
M_{IF}	7	8	8	9	10
Error	9.221E-08	4.560E-07	7.700E-07	6.934E-08	1.689E-07
b0	2.70423445E+00	2.90120692E+00	3.08003115E+00	3.24245257E+00	3.38947677E+00
b1	-8.48937373E-01	-9.06039802E-01	-9.48885247E-01	-9.79961135E-01	-1.00086978E+00
b2	-1.79260656E-03	-4.36735228E-02	-8.70560698E-02	-1.29658962E-01	-1.70496536E-01
b3	-1.48024221E-02	-1.59092479E-02	-2.10651992E-02	-2.99901166E-02	-4.11573849E-02
b4	1.76978730E-02	1.79593944E-02	1.72685505E-02	1.59380192E-02	1.26312285E-02
b5	-7.96026931E-03	-5.49115431E-03	-3.04984781E-03	-1.82211074E-03	1.14612814E-04
b6	3.67757488E-03	4.72086589E-03	4.04045294E-03	5.95050121E-03	6.02343673E-03
b7		-2.16999390E-03	-1.26821254E-03	-3.60390312E-03	-1.71559447E-03
b8				1.92142184E-03	1.18274238E-03
b9					-4.51112565E-04

Forward filter:	**Helmholtz**				
δ	1.4	1.5	1.6	1.7	1.8
$k_f h$	0.710	0.665	0.625	0.584	0.553
η	0.694	0.711	0.727	0.740	0.753
$\hat{G}_d(\pi)$	5.069E-02	4.499E-02	3.905E-02	3.464E-02	3.155E-02
M_{FF}	10	11	12	12	13
Error	2.570E-07	2.573E-07	1.048E-07	2.241E-07	1.937E-07
g0	3.06364076E-01	2.89004418E-01	2.73453343E-01	2.59506668E-01	2.46864255E-01
g1	1.80566354E-01	1.76168717E-01	1.71708869E-01	1.67299074E-01	1.62933860E-01
g2	8.36886435E-02	8.61919769E-02	8.80353631E-02	8.93838791E-02	9.02626280E-02
g3	4.29072252E-02	4.60023015E-02	4.87082106E-02	5.10766140E-02	5.31026669E-02
g4	2.00052156E-02	2.27131028E-02	2.52422738E-02	2.76245331E-02	2.97912104E-02
g5	1.04594277E-02	1.22592591E-02	1.40451696E-02	1.58329180E-02	1.75409055E-02
g6	4.71137920E-03	5.92663808E-03	7.17449792E-03	8.49203227E-03	9.78775081E-03
g7	2.64973884E-03	3.34549853E-03	4.11368138E-03	4.96922605E-03	5.84508755E-03
g8	1.08397918E-03	1.52822407E-03	2.02309155E-03	2.60407509E-03	3.20925734E-03
g9	7.45998391E-04	9.76096668E-04	1.24578267E-03	1.60545074E-03	1.98461462E-03
g10		3.85976013E-04	5.61527866E-04	8.01245246E-04	1.05357758E-03
g11			4.14861310E-04	5.57618588E-04	7.05305331E-04
g12					3.51008657E-04
Inverse filter: N=5					
δ	1.4	1.5	1.6	1.7	1.8
$k_f h$	2.045	1.906	1.789	1.681	1.587
η	0.326	0.355	0.381	0.406	0.429
$\hat{V}_d^N \hat{G}_d(\pi)$	2.673E-01	2.418E-01	2.122E-01	1.900E-01	1.753E-01
M_{IF}	10	11	12	12	13
Error	3.384E-07	9.546E-08	1.018E-07	2.007E-07	8.861E-08
b0	3.52330948E+00	3.64521063E+00	3.75640356E+00	3.85806431E+00	3.95147618E+00
b1	-1.01432707E+00	-1.02164170E+00	-1.02383860E+00	-1.02221004E+00	-1.01756534E+00
b2	-2.08513862E-01	-2.43297739E-01	-2.75029320E-01	-3.03588211E-01	-3.28994190E-01
b3	-5.46436078E-02	-6.96658104E-02	-8.52452627E-02	-1.01175638E-01	-1.17261925E-01
b4	8.19883993E-03	2.64405792E-03	-4.40678370E-03	-1.23155079E-02	-2.08539014E-02
b5	1.00354217E-03	7.88048978E-04	2.05726755E-04	-1.39383477E-03	-3.95369072E-03
b6	6.39448040E-03	7.22229758E-03	7.29925098E-03	6.97925626E-03	6.64669004E-03
b7	-6.08709600E-04	-4.45132448E-04	7.24061411E-04	1.65492275E-03	1.94400154E-03
b8	5.18113491E-04	2.67501201E-03	2.41702916E-03	2.66098154E-03	3.59326458E-03
b9	3.23531373E-04	-2.47727139E-03	-4.20622937E-04	1.60812630E-04	-2.87033778E-04
b10		1.59292430E-03	-3.23120305E-04	-1.02617170E-03	1.56388971E-03
b11			4.15860546E-04	1.22127596E-03	-2.27541893E-03
b12					1.70556098E-03

Forward filter:	**Implicit**				
a	-0.45	-0.4	-0.35	-0.3	-0.25
$k_f h$	0.449	0.638	0.791	0.926	1.043
η	0.813	0.750	0.704	0.667	0.634
$\hat{G}_d(\pi)$	0	0	0	0	0
M_{FF}	14	10	8	7	6
Error	5.417E-07	4.867E-07	5.196E-07	2.945E-07	3.226E-07
g0	1.86693964E-01	2.50108178E-01	2.95951478E-01	3.33365970E-01	3.66180541E-01
g1	1.51835834E-01	1.87548992E-01	2.08357521E-01	2.22336564E-01	2.32085456E-01
g2	9.52244478E-02	9.38588899E-02	8.52018412E-02	7.41067761E-02	6.23087900E-02
g3	5.96836770E-02	4.69245153E-02	3.47860664E-02	2.48056578E-02	1.66956640E-02
g4	3.74655447E-02	2.35464668E-02	1.43211037E-02	8.26309796E-03	4.60093942E-03
g5	2.34778026E-02	1.17671298E-02	5.84185250E-03	2.85777839E-03	1.21887992E-03
g6	1.47719578E-02	5.96684018E-03	2.50131603E-03	9.47140851E-04	
g7	9.25406163E-03	2.97751004E-03	1.01456042E-03		
g8	5.85737779E-03	1.57371401E-03			
g9	3.66736357E-03	7.81853350E-04			
g10	2.35406136E-03				
g11	1.47162302E-03				
g12	9.79628535E-04				
g13	6.09637888E-04				
Inverse filter: N=5					
a	-0.45	-0.4	-0.35	-0.3	-0.25
$k_f h$	1.160	1.524	1.753	1.915	2.045
η	0.595	0.495	0.432	0.386	0.348
$\hat{V}_d^N \hat{G}_d(\pi)$	0	0	0	0	0
M_{IF}	14	10	8	7	6
Error	3.999E-07	7.746E-08	8.612E-08	6.692E-08	7.745E-08
b0	4.20853240E+00	4.06594390E+00	3.78244175E+00	3.56921620E+00	3.38381441E+00
b1	-7.72026844E-01	-1.19718922E+00	-1.25560244E+00	-1.28345962E+00	-1.28245083E+00
b2	-7.37219693E-01	-3.17171841E-01	-1.75268658E-01	-4.78267924E-02	5.14370733E-02
b3	8.90631380E-02	-3.46334470E-02	2.09765559E-02	3.43469752E-02	3.82907321E-02
b4	-3.01424489E-01	-9.79663814E-03	6.81552747E-03	1.28733996E-02	1.54866135E-03
b5	2.00572916E-01	2.45399205E-02	1.49853142E-02	-8.87352951E-04	-7.32841433E-04
b6	-1.83537756E-01	-7.91762352E-03	-6.33498029E-03	3.45290313E-04	
b7	1.60650042E-01	1.41175343E-02	3.20780943E-03		
b8	-1.22905618E-01	-9.73292293E-03			
b9	1.03077153E-01	4.81228612E-03			
b10	-7.33506006E-02				
b11	5.18551199E-02				
b12	-3.11902153E-02				
b13	1.21706469E-02				

Forward filter:	**Implicit**				
a	-0.2	-0.15	-0.1	-0.05	0
$k_f h$	1.155	1.263	1.366	1.470	1.569
η	0.604	0.577	0.550	0.525	0.500
$\hat{G}_d(\pi)$	0	0	0	0	0
M_{FF}	5	5	4	4	2
Error	6.624E-07	5.727E-08	2.204E-07	3.297E-09	3.715E-33
g0	3.95683528E-01	4.23240534E-01	4.49661564E-01	4.74958176E-01	5.00000000E-01
g1	2.39347002E-01	2.44176137E-01	2.47461732E-01	2.49372644E-01	2.50000000E-01
g2	4.99447270E-02	3.74876932E-02	2.51692179E-02	1.25209118E-02	
g3	1.06529981E-02	5.82386289E-03	2.53826796E-03	6.27355538E-04	
g4	2.21350918E-03	8.92039582E-04			
Inverse filter: N =5					
a	-0.2	-0.15	-0.1	-0.05	0
$k_f h$	2.157	2.247	2.328	2.400	2.467
η	0.319	0.291	0.267	0.245	0.226
$\hat{V}_d^N \hat{G}_d(\pi)$	0	0	0	0	0
M_{IF}	5	5	4	4	5
Error	3.181E-07	8.708E-09	2.098E-06	2.315E-06	1.667E-07
b0	3.22526517E+00	3.08128188E+00	2.94435695E+00	2.81439403E+00	2.70353110E+00
b1	-1.27269098E+00	-1.25214599E+00	-1.22255440E+00	-1.18360797E+00	-1.15489889E+00
b2	1.40273950E-01	2.14489036E-01	2.77821526E-01	3.26111295E-01	3.79701344E-01
b3	2.31096601E-02	2.71750113E-03	-2.74456022E-02	-4.97003369E-02	-8.82848484E-02
b4	-3.32521583E-03	-5.70148312E-03			1.17168466E-02

Forward filter:	**Implicit**				
a	0.05	0.1	0.15	0.2	0.25
$k_f h$	1.667	1.771	1.874	1.982	2.094
η	0.475	0.450	0.423	0.396	0.366
$\hat{G}_d(\pi)$	0	0	0	0	0
M_{FF}	4	4	5	5	6
Error	3.297E-09	2.204E-07	5.727E-08	6.624E-07	3.220E-07
g0	5.25041824E-01	5.50338436E-01	5.76759466E-01	6.04316547E-01	6.33835800E-01
g1	2.49372644E-01	2.47461732E-01	2.44176137E-01	2.39346712E-01	2.32081749E-01
g2	-1.25209119E-02	-2.51692179E-02	-3.74876932E-02	-4.99449354E-02	-6.23158421E-02
g3	6.27355538E-04	2.53826796E-03	5.82386289E-03	1.06532876E-02	1.66914416E-02
g4			-8.92039587E-04	-2.21333794E-03	-4.60205778E-03
g5					1.22680945E-03
Inverse filter: N =5					
a	0.05	0.1	0.15	0.2	0.25
$k_f h$	2.526	2.580	2.634	2.683	2.737
η	0.207	0.189	0.172	0.155	0.138
$\hat{V}_d^N \hat{G}_d(\pi)$	0	0	0	0	0
M_{IF}	5	6	6	7	8
Error	3.227E-06	7.413E-07	5.285E-06	2.293E-06	1.944E-06
b0	2.57791721E+00	2.47475000E+00	2.35178325E+00	2.24573892E+00	2.12933320E+00
b1	-1.10352543E+00	-1.06633472E+00	-1.00516102E+00	-9.56688708E-01	-8.92993156E-01
b2	4.10110978E-01	4.51358201E-01	4.67960381E-01	4.93537902E-01	5.05083973E-01
b3	-1.15862247E-01	-1.57912859E-01	-1.86540018E-01	-2.28798644E-01	-2.63483512E-01
b4	2.03180918E-02	4.33344593E-02	6.12034643E-02	9.30687278E-02	1.26598737E-01
b5		-7.82008352E-03	-1.33544314E-02	-3.13156531E-02	-5.46058966E-02
b6				7.32691393E-03	1.99167042E-02
b7					-5.18345153E-03

Forward filter:	**Implicit**			
a	0.3	0.35	0.4	0.45
$k_f h$	2.211	2.346	2.499	2.688
η	0.333	0.296	0.250	0.187
$\hat{G}_d(\pi)$	0	0	0	0
M_{FF}	7	8	10	14
Error	2.945E-07	5.197E-07	4.868E-07	5.420E-07
g0	6.66629639E-01	7.04046044E-01	7.49889577E-01	8.13307141E-01
g1	2.22335937E-01	2.08357049E-01	1.87551511E-01	1.51834225E-01
g2	-7.41053271E-02	-8.51994280E-02	-9.38585474E-02	-9.52273727E-02
g3	2.48062385E-02	3.47824916E-02	4.69229420E-02	5.96795968E-02
g4	-8.26275361E-03	-1.43210902E-02	-2.35458677E-02	-3.74605638E-02
g5	2.85782485E-03	5.84524035E-03	1.17690661E-02	2.34802567E-02
g6	-9.46738592E-04	-2.50250403E-03	-5.96754745E-03	-1.47674731E-02
g7		1.01521873E-03	2.97615185E-03	9.25857717E-03
g8			-1.57282579E-03	-5.86321546E-03
g9			7.80328620E-04	3.66106414E-03
g10				-2.35884643E-03
g11				1.47189839E-03
g12				-9.76099121E-04
g13				6.14381785E-04
Inverse filter: N =5				
a	0.3	0.35	0.4	0.45
$k_f h$	2.791	2.845	2.903	2.975
η	0.121	0.102	0.082	0.057
$\hat{V}_d^N \hat{G}_d(\pi)$	0	0	0	0
M_{IF}	9	10	12	17
Error	2.495E-06	4.997E-06	6.808E-06	5.614E-06
b0	2.00402351E+00	1.86532106E+00	1.70687438E+00	1.50607468E+00
b1	-8.18557646E-01	-7.27388487E-01	-6.15247394E-01	-4.59881242E-01
b2	5.04234876E-01	4.87927719E-01	4.51264344E-01	3.74085994E-01
b3	-2.93313914E-01	-3.13647222E-01	-3.22117689E-01	-3.00464595E-01
b4	1.60403802E-01	1.93568389E-01	2.24216467E-01	2.38508414E-01
b5	-8.23274868E-02	-1.13752825E-01	-1.51506984E-01	-1.86897426E-01
b6	3.84228557E-02	6.35647802E-02	9.96332023E-02	1.44703672E-01
b7	-1.53426265E-02	-3.24980588E-02	-6.29368071E-02	-1.10529005E-01
b8	4.46838776E-03	1.42935610E-02	3.82128591E-02	8.32774799E-02
b9		-4.72838550E-03	-2.13740606E-02	-6.17663483E-02
b10			1.04053409E-02	4.49685900E-02
b11			-3.98646956E-03	-3.20862683E-02
b12				2.21007872E-02
b13				-1.47851853E-02
b14				8.63434275E-03
b15				-5.07933156E-03
b16				2.17278306E-03