Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 206 (2005) 505–535

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A windowing method for periodic inflow/outflow boundary treatment of non-periodic flows

P. Schlatter *, N.A. Adams [1], L. Kleiser

Institute of Fluid Dynamics, ETH Zürich, Sonneggstrasse 3, CH-8092 Zürich, Switzerland

Received 3 August 2004; received in revised form 6 December 2004; accepted 10 December 2004

Abstract

An inflow/outflow boundary treatment procedure is described for the numerical computation of non-periodic flows which allows for the use of periodic spatial boundary conditions. Due to this periodicity, e.g. efficient and accurate Fourier spectral methods can be applied. The governing equations of the flow are modified using window functions as known from signal processing. Thereby, the windowed solution is forced to zero to high order at the artificial boundaries. The physical solution near the boundaries is obtained by a regularised dewindowing operation and boundary conditions are imposed with the help of a suitable base flow which needs to be defined only within the window-boundary regions. On the inner domain, the unmodified flow equations are solved. The base flow can contain spatially and temporally varying disturbances. Hence it is possible to employ transitional and turbulent inflow conditions using the windowing technique.

By properly designing the window function, spectral accuracy of a Fourier discretisation can be obtained. The performance of this scheme is analysed theoretically, verified numerically and compared to the more widely used fringe region technique. It is found that the accuracy of imposing the boundary conditions is similar for both techniques. Furthermore, for flow problems with a spatially evolving base flow, the windowing method does not require the base flow to be periodic.

In this paper, the implementation of the windowing method in a two-dimensional incompressible Navier–Stokes code is examined and compared in detail to the fringe region technique for two test cases: The convection of a localised disturbance and a stationary, spatially evolving jet.

PACS: 47.11.+j; 47.27.Eq; 02.70.Hm

Keywords: Windowing; Navier–Stokes equations; Boundary condition; Periodicity; Fourier discretisation; Fringe method; Incompressible flow

* Corresponding author. Tel.: +41 1 632 0396.
E-mail address: schlatter@ifd.mavt.ethz.ch (P. Schlatter).

[1] Present address: Institute of Aerodynamics, Technical University of Munich, D-85747 Garching, Germany.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.12.015

1. Introduction

For the numerical modelling of a flow problem in an Eulerian description the computational domain is usually a truncation of the real (physical) domain. At the computational domain boundaries the solution is often (partially) unknown, resulting in the need of *artificial boundary conditions* (see the reviews by Givoli [12] and Colonius [9]). The prescription of suitable conditions at an artificial boundary requires that some properties of the solution are known a priori. Since in most cases these properties are known only approximately or can only be guessed, differences between the approximate conditions and the proper physical solution are present near the boundaries. These regions, which are characterised by the changeover from the physical solution to the solution imposed by the (approximate) boundary condition, are commonly considered as "non-physical". At the artificial boundaries also the spatial discretisation has to be constructed such that the resulting numerical scheme is stable and reflections from the boundaries are suppressed.

For finite-difference schemes one usually needs to employ special boundary closures. This is avoided if periodic boundary conditions can be imposed at the artificial boundaries. Then the schemes used in the exterior of the domain can be extended across the artificial boundary and the solution algorithms usually gain robustness and efficiency. Since for the above reasons non-physical regions near artificial boundaries are practically unavoidable, one can exploit this fact by modifying the governing equations in such a way that near the artificial boundaries periodic boundary conditions can be assumed for the discretised spatial-derivative operators of the underlying flow equations without significantly increasing the size of the computational domain. The procedure has to ensure that away from the artificial boundaries the correct physical solution is recovered.

Reducing the effect of artificial boundary conditions by use of a forcing term was first applied by Israeli and Orszag [17]. They used a *sponge layer* to damp oscillations in the solution near the boundaries before being treated by the artificial boundary conditions. In the *perfectly matched layer* (PML) approach [1], it is ensured that the damping layer is non-reflecting by changing the phase speed of the incoming waves such that they are damped. The PML approach has been extended to the treatment of non-uniform Euler flows (see e.g. [15]).

The idea of obtaining non-periodic solutions on periodic domains is due to Spalart [27], who modified the Navier–Stokes equations in "fringe regions" at the computational domain boundaries such that the solution was forced towards periodicity in these regions without affecting the physically meaningful solution in the valuable part of the domain. Spalart demonstrated that accurate results using the fringe method could be obtained. Some justification of the method was also given. The fringe method has found widespread use and was employed successfully in a number of publications involving transitional and turbulent flows, see e.g. [2,19], and the references therein. The fringe method was further analysed in more detail by Nordström et al. [20]. These authors gave a justification of the method and showed quantitatively that for incompressible flow the inflow/outflow problem on the example of the spatially evolving Blasius profile can be simulated with high accuracy.

Here, we adopt a different approach, which was proposed for the simulation of transitional compressible boundary layer flow by Guo et al. [14]. The underlying idea is related to that of the spectral multidomain technique of Israeli et al. [18] and is based on a windowing operation used for the spectral analysis of non-periodic data in signal processing (e.g. [21]). Colonius and Ran [10] used a related approach to simulate flows on unbounded domains by employing a super-grid scale model to stretch the grid near the domain boundaries. In the present work, the method of Guo et al. [14] is revisited and modified with respect to the treatment of the solution near the boundaries. It is then applied to incompressible flows and an extended analysis of its properties is performed.

The paper is organised as follows. Section 2 introduces the fundamentals of both the fringe and the windowing method and their application to flow problems. Section 3 describes a model problem to demonstrate the convergence properties of the windowing technique. In Sections 4 and 5, two numerical test cases involving the Euler and Navier–Stokes equations are presented. Test case 1 examines the ability of both methods to damp out a single localised disturbance as it travels out of the physical domain. Test case 2 considers a spatially evolving jet and assesses the performance of the fringe and the windowing method in simulating a spatially evolving flow.

2. Mathematical formulation

2.1. Problem description and governing equations

We consider incompressible fluid flow on a rectangular two-dimensional *physical domain* Γ_I with boundary $\partial\Gamma_I$. A Cartesian coordinate system is defined such that the streamwise (x_1 or x) and spanwise/cross-stream (x_2 or y) axes are aligned with the straight domain boundaries. The non-dimensional velocity vector with components $u_1 = u$ and $u_2 = v$ is denoted by $\mathbf{u}$. The non-dimensional density ρ and the non-dimensional dynamic viscosity μ are assumed to be constant. Non-dimensionalisation is such that the Reynolds number is given by $Re = \rho/\mu$. The flow is then governed by the incompressible Navier–Stokes equations

$$\partial_t u_i + u_j \partial_j u_i + \frac{1}{\rho}\partial_i p = \frac{1}{\rho}\partial_j \mu(\partial_i u_j + \partial_j u_i), \tag{1}$$

and the continuity equation

$$\partial_k u_k = 0. \tag{2}$$

Here ∂_i and ∂_t denote the partial derivatives with respect to x_i and time t, respectively. Unless stated otherwise, the summation convention over repeated indices applies.

Parts of the boundary $\partial\Gamma_I$ of the physical domain Γ_I can coincide with physical boundaries (e.g. walls), the remainder coincides with artificial boundaries (in particular inflow/outflow boundaries). For notational simplicity we assume that the entire boundary $\partial\Gamma_I$ is artificial. In an application where this is not the case (e.g. at solid walls) the boundary $\partial\Gamma_I$ has to be split accordingly and the following applies to the artificial parts of $\partial\Gamma_I$. Note that for well-posedness of the original problem formulation boundary conditions on the entire boundary $\partial\Gamma_I$ are necessary.

For the following considerations, periodic boundary conditions are assumed in the spanwise/cross-stream direction. In the streamwise direction x the flow is assumed to be spatially evolving. Therefore, the inflow/outflow boundary treatment is imposed in the streamwise direction only. The subsequently described boundary treatment transfers directly to problems in three spatial dimensions and to periodic boundary treatment in more than one direction, but for simplicity we consider here only the two-dimensional case with one inhomogeneous direction.

For the boundary treatment, the physical domain Γ_I is enlarged in the streamwise direction forming the *computational domain* Γ (Fig. 1). On Γ_I we would like to recover an accurate physical solution to the original flow problem according to Eq. (1). The added subdomain $\Gamma_A = \Gamma\backslash\Gamma_I$, on the other hand, is responsible for the boundary treatment and is usually called fringe region. Here, we denote this region – to avoid confusion with the fringe region technique introduced in Section 2.2 – more generally *artificial boundary region*. The computational solution within Γ_A is not part of the physically relevant solution to the original flow problem. The boundary treatment thus modifies the underlying initial-boundary value problem (IBVP) on Γ_I to an IBVP with periodic boundary conditions in the x-direction on the enlarged domain Γ.

For a unique solution of the periodic problem both the fringe and the windowing approach require distributed boundary data to be provided within the artificial boundary region Γ_A. The boundary conditions to the computational problem are thus assumed to be given on the subdomain Γ_A in the form of *distributed boundary conditions* $\mathcal{U}_i$,

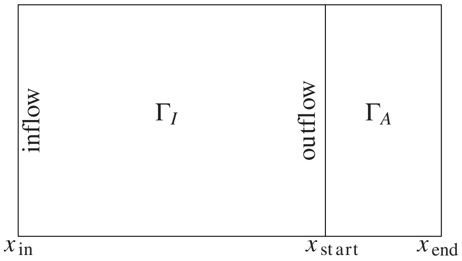

Fig. 1. Sketch of the computational domain Γ consisting of the physical domain Γ_I and the artificial boundary region Γ_A with $\Gamma = \Gamma_I \cup \Gamma_A$. The mean flow velocity is assumed to be from left to right. Periodic boundary conditions are applied in all directions.

$$u_i(\mathbf{x},t) \rightarrow \mathcal{U}_i(\mathbf{x},t) \quad \text{for } x \rightarrow \partial\Gamma_A. \tag{3}$$

The function $\mathcal{U}_i$ needs to be defined on Γ_A only and is usually referred to as (computational) base flow of the problem. $\mathcal{U}_i$ is allowed to vary both in space and time.

2.2. Fringe region technique

The following description of the fringe region technique is based on Lundbladh et al. [19]. The same form has been analysed in [20] and has been successfully applied in a number of cases, e.g. [4].

In order to be able to assume periodicity within the computational domain, the *fringe region* Γ_A is appended downstream of the physical domain Γ_I (see Fig. 1). Within the fringe region, the flow is forced back to the desired inflow condition and possible disturbances are damped by adding a suitable volume force G_i to the right-hand side of the Navier–Stokes equation (1) which vanishes within Γ_I.

The general form of the fringe forcing is given by

$$G_i = \lambda(x)(\mathcal{U}_i - u_i). \tag{4}$$

The fringe function $\lambda(x) \geqslant 0$ is non-vanishing only within the fringe region Γ_A, defined to extend from x_{start} to x_{end}. $\mathcal{U}_i$ is a prescribed flow field, periodic in x, referred to as (computational) base flow, containing the inflow and outflow conditions (see Fig. 2).

If the physical base flow evolves in the streamwise direction x, the streamwise velocity component $\mathcal{U}_x$ of the computational base flow is a blending between inflow and outflow velocity and can be written as [19]

$$\mathcal{U}_x(x,y) = U(x,y) + [U(x - L_P, y) - U(x,y)]S\left(\frac{x - x_{\text{mix}}}{\Delta_{\text{mix}}}\right). \tag{5}$$

Here $U(x,y)$ describes a solution to the Navier–Stokes (or boundary layer) equations in the absence of the periodic boundary treatment, e.g. the Blasius solution in the case of boundary layer flow. $L_P = x_{\text{end}} - x_{\text{in}}$ denotes the length of the periodic computational domain and the smooth step function $S(x)$ is given in Eq. (7). x_{mix} and Δ_{mix} define the properties of the blending and are explained in the following. Note that the blending (5) is only needed for a physical base flow U that is evolving in space, e.g. the simulation of spatially evolving boundary layers. For a base flow that is not dependent on x, $\mathcal{U} = U$ holds.

In two-dimensional flows, the other non-vanishing velocity component $\mathcal{U}_y$ of the base flow can be calculated using the continuity equation, especially in the blending region where the base flow is non-physically modified and $\mathcal{U}_y$ cannot be recovered otherwise. A natural choice for the parameters of the blending region for the coordinates introduced above is $x_{\text{mix}} = x_{\text{start}}$ and $\Delta_{\text{mix}} = x_{\text{end}} - x_{\text{start}}$. This choice ensures that the blending uses the maximum streamwise extent of the fringe region which is preferred for laminar or nearly laminar flows [19]. Hereby it is assured that the effects on the valuable part of the flow domain are minimised.

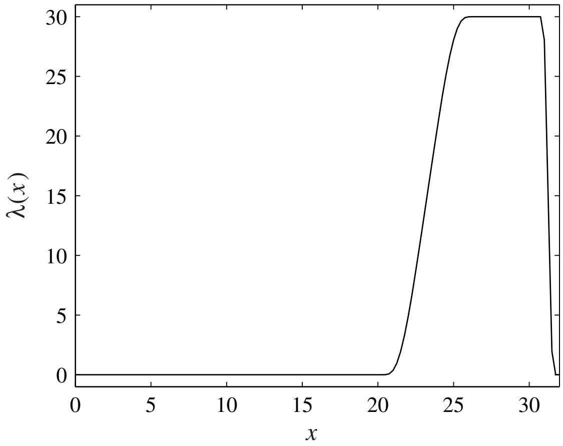

Fig. 2. Sample fringe function $\lambda(x)$ used in test case 2 (Section 5.2) with parameters $\lambda_{\max}$ = 30, x_{start} = 20, x_{end} = 32. Δ_{rise} = 60% and Δ_{fall} = 10% of fringe extent.

It is well possible to include temporally and/or spatially varying inflow disturbances into $\mathscr{U}_i$. Even completely turbulent inflow conditions can be employed in the fringe region, e.g. [4]. Note that these superimposed disturbances should satisfy continuity in such a way that the entire base flow $\mathscr{U}_i$ is divergence-free (see also Section 5.2.1).

The form of the fringe function λ introduced in [19] is

$$\lambda(x) = \lambda_{\max}\left[S\left(\frac{x - x_{\text{start}}}{\Delta_{\text{rise}}}\right) - S\left(\frac{x - x_{\text{end}}}{\Delta_{\text{fall}}} + 1\right)\right]. \tag{6}$$

The maximum strength of the fringe function is $\lambda_{\max}$ and its shape is defined by the function $S(x)$ and the parameters Δ_{rise} and Δ_{fall}. $S(x)$ is a smooth step function with $S(x) = 0$ for $x < 0$ and $S(x) = 1$ for $x \geqslant 1$. The following form of S has continuous derivatives of all orders for $x \neq 0$ and $x \neq 1$

$$S(x) = \begin{cases} 0, & x \leqslant 0, \\ 1/[1 + \exp(\frac{1}{x-1} + \frac{1}{x})], & 0 < x < 1, \\ 1, & x \geqslant 1. \end{cases} \tag{7}$$

Note that the superposition given in Eq. (5) is continuously differentiable only for $x \neq x_{\text{mix}}$ and $x \neq x_{\text{mix}} + \Delta_{\text{mix}}$.

The application of the fringe method imposes an additional restriction on the maximum possible time step of the integration scheme. Straightforward analysis of the linear temporal stability characteristics of the damping term G_i yields the condition

$$\lambda_{\max}\Delta t \leqslant 2.51 \quad \text{resp.} \quad \lambda_{\max}\Delta t \leqslant 2.78 \tag{8}$$

for a third and a fourth order Runge–Kutta scheme, respectively.

2.3. Windowing approach

2.3.1. Theoretical consideration

The windowing method has its roots in signal processing, where the windowing operation allows the spectral analysis of non-periodic signals [21,16]. Similarly, the windowing operation in the present context can be understood as an artificial window through which the physical flow field is projected onto a computational domain.

As a window function on the domain $x \in [x_L, x_R]$ we define a function $w(x) \in C^\infty$ on $\mathbb{R}$ with the following requirements ($x_L < x_l < x_r < x_R$)

- **(R1)** $0 \leqslant w(x) \leqslant 1$ on the entire real axis $\mathbb{R}$,
- **(R2)** $\max\{1 - w(x)\} \leqslant \epsilon_2$ on the inner domain $[x_l, x_r]$,
- **(R3)** $w(x) \leqslant \epsilon_1 \mathrm{e}^{-\alpha|x-x_L|}$ for $x < x_l$, $w(x) \leqslant \epsilon_1 \mathrm{e}^{-\alpha|x-x_R|}$ for $x > x_r$ with some $\alpha > 0$, i.e. $w(x)$ decays at least exponentially for $x \to \pm\infty$. An implication of this requirement is $\max\{w(x_L), w(x_R)\} \leqslant \epsilon_1$.

Any bounded and continuous function $f(x)$ defined on $\mathbb{R}$ has a Fourier transform if it is multiplied with such a window function $w(x)$. Accordingly, we define the windowing operation by

$$\left|\tilde{f}(x)\right| := w(x)f(x). \tag{9}$$

The Fourier transform of $\tilde{f}$ is then defined as

$$\hat{\tilde{f}}(k) = \int_{-\infty}^{\infty} \tilde{f} \mathrm{e}^{-\mathrm{i}kx}\, \mathrm{d}x, \tag{10}$$

and for $\tilde{f}^{(p)}$ integrable and $\tilde{f} \in C^{p-1}$ one can show by partial integration that

$$\hat{\tilde{f}}(k) = \mathcal{O}(|k|^{-p}). \tag{11}$$

For $\tilde{f} \in C^\infty$ the inverse transform of $\hat{\tilde{f}}$ converges to $\tilde{f}$ spectrally.

The error introduced by performing the Fourier integral (10) only over a bounded domain $[x_L, x_R]$ according to

$$\hat{\tilde{f}}_\delta(k) = \int_{x_L}^{x_R} \tilde{f}(x) \mathrm{e}^{-\mathrm{i}kx}\, \mathrm{d}x, \tag{12}$$

instead of $(-\infty, +\infty)$ is estimated with **(R3)** as

$$\epsilon'_\delta = \max\left|\hat{\tilde{f}} - \hat{\tilde{f}}_\delta\right| \leqslant \left|\int_{-\infty}^{x_L} \tilde{f} \mathrm{e}^{-\mathrm{i}kx}\, \mathrm{d}x\right| + \left|\int_{x_R}^{\infty} \tilde{f} \mathrm{e}^{-\mathrm{i}kx}\, \mathrm{d}x\right| \leqslant 2\frac{M\epsilon_1}{\alpha} \tag{13}$$

with $M \geqslant \max |f(x)|$. Considering the band-limited inverse transform of Eq. (10)

$$\tilde{f}_N = \frac{1}{2\pi} \int_{-k_N}^{k_N} \hat{\tilde{f}}(k) \mathrm{e}^{\mathrm{i}kx}\, \mathrm{d}k, \tag{14}$$

where the cut-off wavenumber k_N is the Nyquist wavenumber $k_N = \pi/h$ with the grid spacing h, the estimate for the *truncation error*

$$\epsilon_N = \max\left|\tilde{f}_N - \tilde{f}\right| \leqslant \left|\frac{1}{2\pi}\left(\int_{-\infty}^{-k_N} + \int_{k_N}^{\infty}\right) \hat{\tilde{f}}(k) \mathrm{e}^{\mathrm{i}kx}\, \mathrm{d}k\right| \leqslant C|k_N|^{1-p} \tag{15}$$

with a constant C decays at least as $\mathcal{O}(|k_N|^{1-p})$. For band-limited $\tilde{f}_N$, it can be shown using Whittaker cardinal functions that the Fourier transform of $\tilde{f}_N$ is given as [3]

$$\hat{\tilde{f}}_N(k) = h \sum_{j=-\infty}^{\infty} \tilde{f}(x_j) \mathrm{e}^{-\mathrm{i}kx_j} \quad \text{with } x_j = j \cdot h;\ h = \frac{\pi}{k_N}. \tag{16}$$

By a similar procedure as for the relation (13), the window truncation is imposed on $\hat{\tilde{f}}_N$ by terminating the summation in Eq. (16) at some integer lower and upper bounds j_L and j_R, respectively, where $x_L \leqslant x_{j_L} < x_L + h$ and $x_R - h < x_{j_R} \leqslant x_R$. For the resulting $\hat{\tilde{f}}_{N_\delta}$ one arrives at the estimate

$$\epsilon''_\delta = \max\left|\hat{\tilde{f}}_N - \hat{\tilde{f}}_{N_\delta}\right| \leqslant h\left|\sum_{j=-\infty}^{j_L} \tilde{f}(x_j)\mathrm{e}^{-\mathrm{i}kx_j}\right| + h\left|\sum_{j=j_R}^{\infty} \tilde{f}(x_j)\mathrm{e}^{-\mathrm{i}kx_j}\right| \leqslant 2\epsilon_1 hM\left(1 + \mathrm{e}^{-\alpha h}\right), \tag{17}$$

which is larger than ϵ'_d of Eq. (13) only for $(\alpha h) \gtrsim 0.6590$, i.e. for coarse grids. The error $|\tilde{f}_{N_\delta} - \tilde{f}_N|$ can be estimated by taking the inverse Fourier transform of $(\hat{\tilde{f}}_N - \hat{\tilde{f}}_{N_\delta})$ and using Eq. (17) as

$$\epsilon_\delta = \max\left|\tilde{f}_{N_\delta} - \tilde{f}_N\right| \leqslant 2\epsilon_1 M\left(1 + \mathrm{e}^{-\alpha h}\right) \leqslant 4\epsilon_1 M \tag{18}$$

and is usually referred to as *window truncation error*. The overall error ϵ_w is then composed of both the truncation error and the window truncation error as

$$\epsilon_w = \max\left|\tilde{f}_{N_\delta} - \tilde{f}\right| \leqslant \epsilon_\delta + \epsilon_N. \tag{19}$$

For $f(x) \in C^\infty$ in particular this means that the error ϵ_w decays exponentially until the window truncation error ϵ_δ is reached. ϵ_δ can be made as small as the machine precision by adjusting the value of ϵ_1 accordingly (see Eq. (13)).

For functions which are defined on a discrete grid, analogous derivations and estimates hold. In this case the Fourier transforms are replaced by discrete transforms and the integrals are replaced by summations using the trapezoidal rule.

The inverse windowing process is singular or at least ill-posed on the domain boundaries $x = x_L$ and $x = x_R$ since there $w \approx 0$. Retrieving the quantity f from the windowed value $\tilde{f}$ is ill-conditioned for any $x \in [x_L, x_l]$ and $x \in [x_r, x_R]$ for which w is small. Therefore, a regularised *de-windowing operation* is introduced as

$$\overline{f} = (1 - w)\mathscr{F} + \tilde{f}. \tag{20}$$

Here $\mathscr{F}$ is a function defined at least on $x \in [x_L, x_l]$ and $x \in [x_r, x_R]$ describing the distributed boundary data required for the solution of the periodic IBVP. In the context of a flow simulation, $\mathscr{F}$ is the base flow. The maximum regularisation error $|f - \overline{f}|$ on $x \in [x_l, x_r]$ is bounded by $(\epsilon_2 \cdot \max|\mathscr{F} - f|)$, whereas on $x \in [x_L, x_l]$ and $x \in [x_r, x_R]$ it is bounded by $\max|\mathscr{F} - f|$.

Different types of window functions can be designed. Since no window function used in signal processing (see [16]) directly suits the needs for the present application, different approaches were studied in [13]. A window function satisfying the requirements **(R1)**–**(R3)** is

$$w(x) = 10^{-a^n|2(x-x_L)/(x_R-x_L)-1|^n}, \tag{21}$$

which assumes that the window function describes a symmetric window ($x_l + x_r = x_L + x_R$). We refer to the window function (21) as an *exponential window* since it satisfies requirement **(R3)** and thus preserves spectral convergence of the Fourier series of a sufficiently smooth windowed function according to the above derivation. The parameters a and n can be calculated from the conditions that for some $x_L < x_l < x_r < x_R$

$$w(x_L) = w(x_R) \leqslant \epsilon_1 \quad \text{and} \quad w(x_l) = w(x_r) \geqslant 1 - \epsilon_2 \tag{22}$$

with small numbers ϵ_1 and ϵ_2, e.g. comparable to the machine precision. Analytical relations for the derivatives $\partial_x w$ and $\partial_{xx} w$ can be derived easily. If desired, an extension to non-symmetric windows is straightforward by connecting two windows according to Eq. (21) with different parameters a and n in the middle of the domain $x = \frac{1}{2}(x_L + x_R)$.

An example of an exponential window function is shown in Fig. 3 together with the spectrum of the windowed non-periodic function introduced in [14] (scaled according to $x' = (x - x_L)/(x_R - x_L)$)

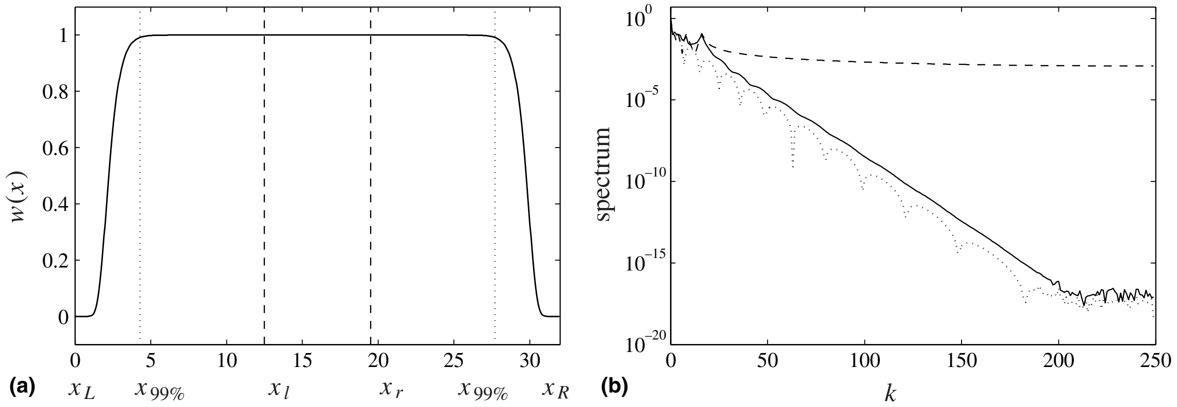

Fig. 3. *Left:* Window function used in test case 2 (Section 5.2). Parameters are $n = 27$, $a = 1.111$, $x_L = 0$, $x_R = 32$, $x_l = 12.5$, $x_r = 19.5$ (vertical dashed lines), $w(x) = 0.99$ at $x = 4.3$ and $x = 27.7$ (vertical dotted lines), spectral bandwidth $k_w \approx 200$. *Right:* Spectra of ··· $w(x)$, ---- $f(x)$ (Eq. (23)), — $f(x)w(x)$.

$$f(x') = \tanh(4x') + \mathrm{e}^{-4x'} \sum_{k=0}^{4} \sin(2\pi 2^k x'), \quad 0 \leqslant x' \leqslant 1. \tag{23}$$

It is evident from Fig. 3(b) that this particular choice of the window function $w(x)$ preserves the spectral convergence of the Fourier representation of $f(x)w(x)$.

The efficiency of a window function can be estimated by the following consideration: The spectral bandwidth k_w of a window function $w(x)$ is defined by

$$|\hat{w}(k)| < \delta \quad \text{for } |k| \geqslant k_w \tag{24}$$

with $\hat{w}(k)$ denoting the Fourier coefficients according to Eq. (10) and δ a fixed small error level. A harmonic function $g(x)$ with unit amplitude is given as (assuming $x_L = 0$)

$$g(x) = \sin\left(\frac{2\pi}{x_R} k_g x\right). \tag{25}$$

The windowed function $\tilde{g}(x)$ is then, due to the convolution properties of the windowing process in Fourier space, resolved up to the error δ if at least

$$N \geqslant 2(k_w + k_g) \tag{26}$$

grid points are used in the discretisation of $w(x)$ and $g(x)$. For an exponential window the spectral bandwidth k_w is proportional to the physical extent of the domain divided by the extent of the windowing regions

$$k_w \propto \frac{x_R - x_L}{(x_l - x_L) + (x_R - x_r)}. \tag{27}$$

Therefore, k_w is determined by the choice of the computational domain Γ alone. Eq. (26) now shows that the overhead cost of the windowing process, i.e. the fraction k_w/N, will decrease with increasing resolution of the discretisation, i.e. with increasing k_g. The efficiency of the windowing methods lies thus in large N; moreover, the efficiency of a fast Fourier transformation (FFT) possibly used in the numerical algorithm also increases with N.

To apply the windowing procedure in the general case of $D > 1$ dimensions, the window function is extended tensorially by introducing functions $W_i(x_i)$ for each coordinate direction x_i analogously to $w(x)$ of Eq. (9). The window function is then defined as

$$W(\mathbf{x}) = \prod_{i=1}^{D} W_i(x_i) \tag{28}$$

and the windowing operation as

$$\tilde{f}_i(\mathbf{x}) = W(\mathbf{x}) f_i(\mathbf{x}), \quad i = 1, \ldots, D \tag{29}$$

for a D-component vector function $f_i(\mathbf{x})$.

2.3.2. Application to non-periodic flows

We again consider the flow in the domain Γ. Similarly to the fringe method, Γ_I denotes the subdomain of Γ in which we want to recover the physically relevant solution (see sketch Fig. 4). Γ_I consists of the domain Γ_I^I, where a solution of the unmodified equation (1) is sought, and the domain Γ_I^A, where physical correct inflow data are prescribed (distributed inflow conditions). The (normally not exactly known) outflow is treated in Γ_A, which is not part of the physical solution to the problem (distributed outflow conditions). The class of problems that can be treated with the windowing technique is summarised by the following properties:

- (**A1**) There exists a base flow $\mathcal{U}_i$ which is at least defined on Γ_I^A and Γ_A. This base flow is used as distributed inflow and outflow condition and therefore contains the boundary data for the flow problem. $\mathcal{U}_i$ can be constructed similar to Eq. (5) for the fringe method. It should satisfy continuity $\partial_i \mathcal{U}_i = 0$. For parts of Section 4.2 it will further be assumed that $\mathcal{U}_i$ is periodic in the flow direction x.
- (**A2**) The solution on Γ_I is subject to the distributed boundary data

  $$u_i \to \mathcal{U}_i \quad \text{for } x \to \partial\Gamma_A. \tag{30}$$

  On the inflow boundary of the physical domain Γ_I the inflow condition $u_i = \mathcal{U}_i$ on the inflow portion of $\partial\Gamma_I$ is fulfilled exactly.
- (**A3**) The solution u_i is integrated according to the unmodified equation (1) only on Γ_I^I. The known solution on Γ_I^A is expected to provide accurate inflow data (solution to the Navier–Stokes equations). On Γ_A the outflow condition is imposed.

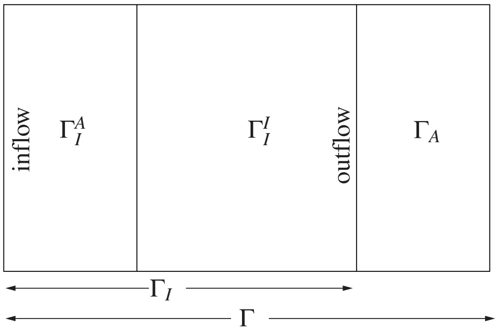

Fig. 4. Sketch of the computational domain Γ, adapted for the windowing technique, consisting of the physical domain Γ_I^I, the inflow domain Γ_I^A and the outflow domain Γ_A. $\Gamma = \Gamma_I \cup \Gamma_A$ with $\Gamma_I = \Gamma_I^I \cup \Gamma_I^A$.

In analogy with the fringe technique, the base flow $\mathcal{U}_i$ on Γ_I^A can e.g. contain superimposed temporally and spatially varying disturbances. Again, it is advisable that these disturbances satisfy continuity, see also Section 5.2. It should be noted that, since the inflow window is located within the physical domain Γ_I, there should be no non-physical energy feed into the flow and the velocities should follow physical evolution equations, e.g. in case of transition simulations results from linear stability theory.

The main difference of (**A1**)–(**A3**) to the properties of the fringe method is that Γ_I^A is assumed to be part of the physical domain Γ_I. This slight modification, however, does not pose a serious restriction on both the generality of problems that can be treated using the windowing method and the validity of the solution. Flow problems that are to be treated with the windowing technique (and similarly with the fringe method) can be distinguished into two categories:

- Damping of outflowing disturbances together with an undisturbed inflow within an otherwise periodic base flow. In this case, perturbations of the base flow are usually introduced in the domain Γ_I^I and the *downstream* evolution is investigated (see test case 1 in Section 5.1), e.g. the flow around bodies or the spatial evolution of turbulent spots in a channel flow. For these cases, the base flow is usually independent of the streamwise variable x.
- A spatially evolving flow is examined. Here, the inflow on Γ_I^A is a valid solution to the Navier–Stokes equations. In this case the starting point of the physical domain is usually a matter of definition or its precise location is not important, e.g. boundary layer simulations starting downstream of the leading edge [4].

2.3.3. Windowed flow equations

In this section, the windowed evolution equation for the velocities u_i and pressure p are derived from the Navier–Stokes equations (1). Recall from Eq. (29) that the windowing operation is defined as

$$\tilde{u}_i(\mathbf{x}) = W(\mathbf{x})u_i(\mathbf{x}), \tag{31}$$

and from Eq. (20) the definition of the de-windowing operation

$$\overline{u}_i = (1-W)\mathcal{U}_i + \tilde{u}_i. \tag{32}$$

As mentioned earlier, the maximum regularisation error on Γ_I^I is bounded by $(\epsilon_2 \cdot \max|\mathcal{U}_i - u_i|)$, whereas on Γ_A and Γ_I^A it is bounded by $\max|\mathcal{U}_i - u_i|$, which is small by (**A2**). The distributed boundary data of the solution are enforced by means of the dewindowing operation (32). From Eq. (31) it follows that

$$\partial_j u_i = \frac{1}{W}(\partial_j \tilde{u}_i - u_i \partial_j W) \tag{33}$$

which is singular on $\partial\Gamma$. Note that in Eq. (33) derivative operations are taken only on windowed variables and the window function. Therefore, it is possible to use schemes assuming periodic boundary conditions to evaluate the derivatives, e.g. accurate global Fourier methods.

Similarly, we obtain for the gradient of the convective fluxes $F_{ij}^c = u_i u_j$

$$\partial_j F_{ij}^c = \frac{1}{W}\left(\partial_j \tilde{F}_{ij}^c - F_{ij}^c \partial_j W\right) = \frac{1}{W}\left(\partial_j \widetilde{u_i u_j} - u_i u_j \partial_j W\right). \tag{34}$$

The diffusive fluxes become

$$F_{ij}^d = \mu(\partial_i u_j + \partial_j u_i) = \frac{1}{W}\mu(\partial_i \tilde{u}_j + \partial_j \tilde{u}_i - u_j \partial_i W - u_i \partial_j W), \tag{35}$$

and the respective windowed flux is

$$\tilde{F}^d_{ij} = \mu(\partial_i\tilde{u}_j + \partial_j\tilde{u}_i - u_j\partial_i W - u_i\partial_j W). \tag{36}$$

On taking the gradient of the diffusive flux one obtains by (33)

$$\partial_j F^d_{ij} = \frac{1}{W}\left(\partial_j\tilde{F}^d_{ij} - F^d_{ij}\partial_j W\right). \tag{37}$$

Note that either form of F^d_{ij} in Eq. (35) can be inserted in (37). Depending on the numerical scheme it might be impractical to compute derivatives $\partial_i\, u_j$ on the non-windowed quantities. On the other hand, it is usually possible – depending on the window function W – to evaluate the term $\partial_j W/W$ despite the singularity of W on $\partial\Gamma$ since $\partial_j W/W$ is bounded; e.g. for the exponential window (21) an analytical expression can be given. The second form in Eq. (35) is preferred in such cases.

On substitution of (34) and (37) into (1) and (2) one can derive the modified Navier–Stokes equations governing the evolution of $\tilde{u}_i$

$$\begin{aligned}\partial_t\tilde{u}_i + \partial_j\widetilde{u_j u_i} + \frac{1}{\rho}\partial_i\tilde{p} - \frac{1}{\rho}\partial_j\left[\mu(\partial_i\tilde{u}_j + \partial_j\tilde{u}_i)\right] = {} & u_i u_j\partial_j W + \frac{p}{\rho}\partial_i W - \frac{1}{\rho}\partial_j\left[\mu(u_j\partial_i W + u_i\partial_j W)\right] \\ & - \frac{1}{\rho}\frac{\partial_j W}{W}\mu(\partial_i\tilde{u}_j + \partial_j\tilde{u}_i - u_j\partial_i W - u_i\partial_j W),\end{aligned} \tag{38}$$

$$\partial_k\tilde{u}_k = u_k\partial_k W, \tag{39}$$

together with the dewindowing operation (32), which enforces the distributed boundary conditions.

The linear stability properties of the temporally discretised equation (38) have been analysed in [13]. There, it is concluded that no additional time step restrictions are encountered.

Further refinements of the scheme just described are possible. Guo [13] proposes for his compressible calculations the introduction of a *sponge region* close to the outflow and a *buffer domain* within the windowing region. However, for the present study, neither of these refinements were found necessary.

2.3.4. Formal comparison

It is instructive to compare the fringe and windowing techniques on a formal level by a simple analogy. Consider the fringe method applied to an evolution equation $\partial_t u_i + F_i(u) = 0$ in the following form (see Section 4.1)

$$\partial_t u_i + F_i(u) = \lambda(\mathscr{U}_i - u_i). \tag{40}$$

Using a fractional step approximation in the framework of an Euler-forward integration, the discrete time step Δt is split into two substeps of length τ and τ'

$$\frac{u'_i - u^{(n)}_i}{\tau} = -F_i(u), \tag{41}$$

$$\frac{u^{(n+1)}_i - u'_i}{\tau'} = \lambda(\mathscr{U}_i - u'_i). \tag{42}$$

The second equation can be rewritten as

$$u^{(n+1)}_i = (1 - \tau'\lambda)u'_i + \tau'\lambda\mathscr{U}_i. \tag{43}$$

The windowing approach enforces the boundary conditions via the dewindowing (32) at the end of each substep. Formally written as

$$u_i^{(n+1)} = (1 - W)\mathcal{U}_i + W\overline{u}_i', \tag{44}$$

where $\overline{u}_i'$ is calculated in a forward integration similar to (41). Comparing Eqs. (43) and (44) yields the result

$$W = 1 - \tau'\lambda. \tag{45}$$

Without going into further details, Eq. (45) gives – except for the fractional step error – a formal framework to compare the fringe and the windowing techniques.

3. Application to an ordinary differential equation

This section presents the application of the windowing technique to an ordinary differential equation. General properties such as convergence and accuracy are discussed and compared to the theoretical results of Section 2.3.

In the following, the one-dimensional Poisson equation, formulated as a simple boundary-value problem,

$$u''(x) = f(x) \tag{46}$$

is considered on the bounded domain $x \in [0,4]$. The prime denotes derivation with respect to x. The right-hand side is set to

$$f(x) = -15\mathrm{e}^{-1-x}\cos(4x) + 8\mathrm{e}^{-1-x}\sin(4x) \tag{47}$$

and the boundary conditions are chosen as

$$u(0) = \frac{1}{e} \quad \text{and} \quad u(4) = \frac{1}{e^5}\cos(16). \tag{48}$$

The analytical solution is non-periodic on [0,4] and given by (see Fig. 5)

$$u_{\mathrm{ex}} = \mathrm{e}^{-1-x}\cos(4x). \tag{49}$$

Using the windowing technique (9) and the relations for the derivatives (33) one finds for the windowed solution $\tilde{u}$

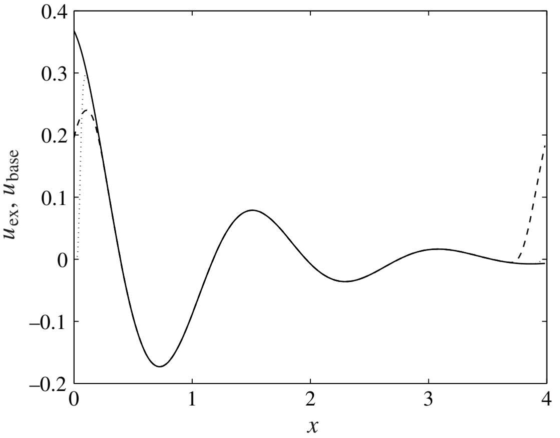

Fig. 5. — Exact solution u_{ex} of Eq. (46) given in (49), the modified periodic base flows with ··· additional window (case 3) and ---- blending region (case 4).

$$\tilde{u}(x) = w(x)u(x), \tag{50}$$

$$\tilde{f}(x) = w(x)f(x), \tag{51}$$

$$\tilde{u}'' = \tilde{f} + uw'' + 2u'w' \tag{52}$$

$$= \tilde{f} + uw'' + 2\frac{w'}{w}(\tilde{u}' - uw'). \tag{53}$$

The regularised dewindowing operation (20) is given as

$$\overline{u} = (1 - w)u_{\text{ex}} + \tilde{u}, \tag{54}$$

which is considered as the numerical solution to Eq. (46) satisfying the boundary conditions (48), which are enforced by the superposition of u_{ex}. To solve Eq. (52) it is necessary to evaluate the term $2u'w'$ which includes a derivative of a non-windowed quantity. Unless u is periodic, the evaluation of u' in Fourier space converges only linearly in the number of Fourier modes. Alternatively, Eq. (53), which is formally identical to (52), can be used. There, only derivatives of windowed quantities and that of $w(x)$ are needed. For the numerical solution, the derivatives are calculated pseudospectrally with Fourier expansion.

Four possibilities to calculate the right-hand side have been evaluated. Firstly, the (usually unknown) exact analytical derivative of u_{ex} is used to evaluate the right-hand side of (52). In the other three cases the derivatives are computed numerically: The second implemented solution algorithm solves Eq. (53) in which all the derivatives are taken from windowed quantities, thus directly allowing the use of Fourier methods. This is the preferred variant due to its generality. The last two options artificially modify the exact solution u_{ex} (boundary conditions) towards a periodic base solution u_{base} in order to evaluate $2u'w'$ and thus solve Eq. (52): Algorithm 3 uses a second window function to enforce periodicity (see Fig. 6). The fourth version replaces the exact solution in the end regions of the domain (0,4) with a weighted superposition of the exact solution $u_{\text{ex}}(x)$ and the solution at the opposite side of the domain $u_{\text{ex}}(x - 4)$ (compare the blending technique in Eq. (5)).

For all variants, the window function $w(x)$ according to Eq. (21) has been chosen with $\epsilon_1 = \epsilon_2 = \epsilon$ being the machine precision (double precision, 2.2×10^{-16}). The particular choice of the parameters of $w(x)$ for variants 3 and 4 ensures that the window function vanishes in the main part of the blending region $(0, 0.4)$ and $(3.6, 4)$. The different window functions are shown in Fig. 6.

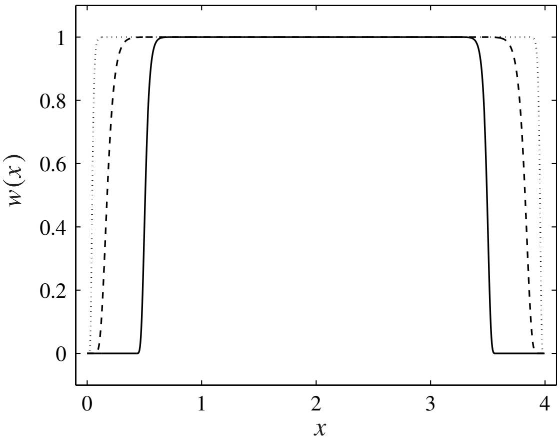

Fig. 6. Different window functions: ---- Main window for cases 1 and 2, $w(x) \leqslant \epsilon$ for $x = 0$ and $x = 4$, $w(x) \geqslant 1 - \epsilon$ for $(1.2, 2.8)$; — main window for cases 3 and 4, $w(x) \leqslant \epsilon$ for $(0.4, 3.6)$, $w(x) \geqslant 1 - \epsilon$ for $(1.2, 2.8)$; ··· additional window for case 3, $w(x) \leqslant \epsilon$ for $x = 0$ and $x = 4$, $w(x) \geqslant 1 - \epsilon$ for $(0.4, 3.6)$.

The convergence rate of the calculated solution (52)–(54) toward the exact solution (49) with increasing number of grid points is plotted in Fig. 7 for all variants of evaluating equations (52) and (53). The error is computed on the entire domain $\Gamma = [0,4]$ (length $L = 4$) and is defined as

$$L_2(u) = \left[\frac{1}{L}\int_0^L |\overline{u} - u_{\mathrm{ex}}|^2\,\mathrm{d}x\right]^{1/2}. \tag{55}$$

As expected, taking the exact solution of the term $2u'w'$ is the most accurate variant (although not feasible in real applications) and thus leading to the best convergence rate. While increasing the number of grid points N, especially the convergence rate of variant 3 (additional window to enforce periodicity) declines significantly. The inclusion of large gradients in the base flow u_{ex} for this variant reduces the smoothness of the solution such that spectral convergence cannot be sustained. The evaluation of variant 2 solving Eq. (53) results in good spectral convergence, although starting at a higher error level for small N. The reason for this behaviour is the numerical accuracy of the term $2\frac{w'}{w}(\tilde{u}' - uw')$. There, the difference of two numbers of the same order is multiplied with the – possibly large but bounded – factor $\frac{w'}{w}$.

Fourier spectra of the solution are shown in Fig. 8. The windowed solution $\tilde{u}$ of variant two shows the same behaviour as seen in Fig. 3: The spectrum of the windowed solution decays exponentially. The spectrum of the (periodic) dewindowed solution $\overline{u}$ of case 3 indicates that the windowing as a means to construct a periodic function preserves spectral accuracy of the solution, although with a slow spectral convergence rate due to the steep gradients. On the other hand, the blending towards a periodic function with the smooth step function $S(x)$ (variant 4) does not result in a spectrally accurate solution $\overline{u}$ because the superposition is not in C^∞ (see comments on Eq. (7)). Nevertheless, as implied by Fig. 7, this does not result in a significant decrease of the spectral convergence coefficient (slope of curves in Fig. 7) of the dewindowed solution for large N.

The example of this section leads to the conclusion that it is possible to use the windowing technique to solve differential equations in Fourier space with given non-periodic distributed boundary conditions. If the window function is chosen according to the properties (**R1**)–(**R3**) (Section 2.3), the windowing technique preserves spectral convergence of the computed and the dewindowed solution $\overline{u}$ towards the exact solution u_{ex}.

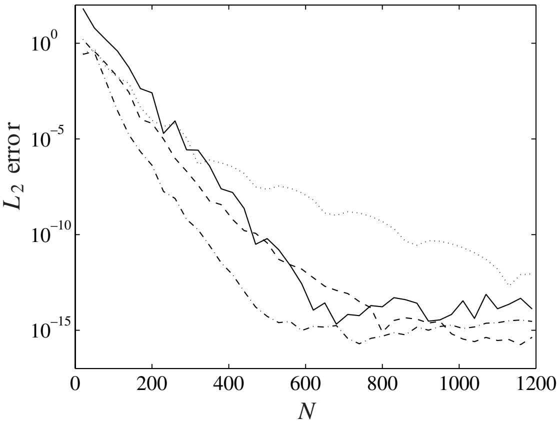

Fig. 7. L_2 error of the numerical solution with increasing number of grid points for the different variants of solving Eq. (52): –·– case 1 (exact derivative), ··· blending with additional window (variant 3), ---- blending with step function (variant 4). Using Eq. (53): — variant 2 (non-periodic base flow).

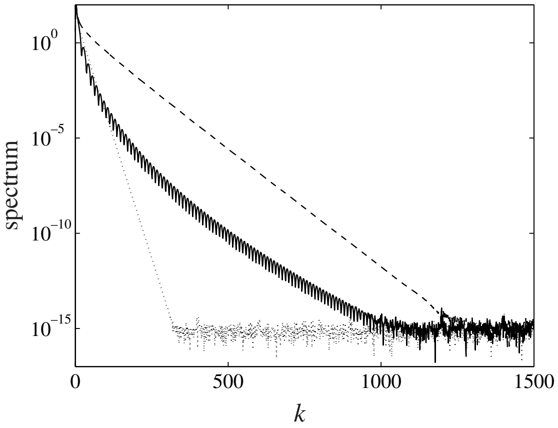

Fig. 8. Fourier spectra of the ··· windowed solution $\tilde{u}$ (variant 2), — the dewindowed solution $\bar{u}$ of variant 4 (step function) and ---- variant 3 (windowing).

Furthermore, different ways to evaluate derivatives of the base solution have been considered. It has been shown that the transformation of the differential equation into a form in which derivatives are applied only to windowed and thus periodic quantities is possible. Moreover, the size of the artificial boundary domain is kept low. On the other hand, the construction of a periodic base solution $\mathcal{U}$ by using a blending similar to that of the fringe technique (5) also proved to be successful. Using a second window to construct a periodic base solution should be avoided since the convergence rate is lower due to the larger spectral bandwidth of the solution caused by the steep window function.

4. Implementation for 2D Navier–Stokes equations

The numerical discretisation scheme employed for the following two test cases solves the incompressible Navier–Stokes equations in two dimensions with a simple and straightforward algorithm. The spatial discretisation uses Fourier decomposition in both directions. Aliasing errors arising from the pseudospectral computation of the nonlinear terms are treated by the 3/2-rule [7]. For time advancement, an explicit third-order Runge–Kutta scheme with adaptive time stepping is employed [26]. The divergence-free condition (2) is enforced by a Helmholtz projection [8]. An 8th order exponential filter as in [11] was used in the Poisson step of the solution algorithm to achieve the stability of the fractional-step method while maintaining sufficient accuracy. The code was validated separately.

For an explicit Runge–Kutta time discretisation, it is sufficient to explain the numerical details of an Euler-forward time step. The updated velocity of a sub-step will be denoted by $u^{(n+1)}$. At time $t^{(n)} = n\tau$ the solution $\mathbf{u}^{(n)}$ on Γ is assumed to satisfy $\nabla \cdot \mathbf{u}^{(n)} = 0$.

4.1. Fringe method

For the fringe method, the forcing term G_i (see Eq. (4)) is evaluated and included into the Navier–Stokes equations according to (1). We proceed as follows:

- **(F1)** Using Eq. (1) and the abbreviation F_i including the nonlinear and viscous terms and the forcing term G_i

$$\partial_t u_i + \frac{1}{\rho}\partial_i p = F_i(\mathbf{u}^{(n)}) \tag{56}$$

we obtain for the first step of the algorithm

$$u_i^* = u_i^{(n)} + \alpha\tau F_i(\mathbf{u}^{(n)}). \tag{57}$$

The step size of the Runge–Kutta step is given by $\alpha\tau$ where τ is the full time-step.

- **(F2)** The divergence-free condition on the intermediate solution **u*** is enforced by a Helmholtz projection

$$u_i^{(n+1)} = u_i^{**} = u_i^* - \partial_i\phi, \tag{58}$$

where ϕ is a scalar function and is determined from

$$\partial_k u_k^* = \partial_k\partial_k\phi. \tag{59}$$

The pressure can in principle be recovered directly from the fractional step algorithm, although the accuracy of the pressure solution is limited to first order in time [6,22].

4.2. Windowing technique

For the windowing technique, Eqs. (38) and (39) have to be solved. Basically, the same algorithm as for the fringe technique is used, although some modifications are appropriate. Instead of using the windowed quantity $\tilde{\mathbf{u}}$ as dependent variable, the regularised dewindowed quantities $\overline{u}_i$ (see Eq. (32)) have been used in the code to minimise the code changes.

Similarly to Section 4.1, for each Runge–Kutta substep we proceed as follows:

- **(W1)** Using Eq. (38) we introduce $\tilde{F}_i$ according to

$$\partial_t\tilde{u}_i + \frac{W}{\rho}\partial_i p = \tilde{F}_i(\overline{\mathbf{u}}^{(n)}). \tag{60}$$

The forward projection is given for the windowed solution $\tilde{\mathbf{u}}$ as

$$\tilde{u}_i^* = \tilde{u}_i^{(n)} + \alpha\tau\tilde{F}_i(\overline{\mathbf{u}}^{(n)}) \tag{61}$$

or using the regularised dewindowed variables due to linearity

$$\overline{u}_i^* = \overline{u}_i^{(n)} + \alpha\tau\tilde{F}_i(\overline{\mathbf{u}}^{(n)}) \tag{62}$$

for a stationary base flow. Alternatively, for non-stationary base flows $\mathcal{U}_i = \mathcal{U}_i(t)$, the time derivative of $\overline{u}_i$ (see Eq. (32)) has to be included, replacing Eq. (62) by

$$\overline{u}_i^* = \overline{u}_i^{(n)} + \alpha\tau\big(\tilde{F}_i(\mathbf{u}^{(n)}) + (1-W)\partial_t\mathcal{U}_i\big). \tag{63}$$

Note that $(1-W)\partial_t\mathcal{U}_i$ can alternatively be added in step **(W3)** since $\mathcal{U}_i$ is divergence-free by assumption. The test cases have shown (see Sections 5.1 and 5.2) that an additional dewindowing operation at this point increases the accuracy of the results slightly

$$\overline{u}_i^{*+} = (1-W)\mathcal{U}_i + W\overline{u}_i^*. \tag{64}$$

The reason for this is that the additional dewindowing enforces the boundary data on the intermediate solution $\overline{u}_i^{*+}$ and damps possible artefacts of $\tilde{F}_i$ in the windowing regions. A similar observation is also made in [10]. These authors found an additional filtering step in the fringe region to be beneficial.

- **(W2)** The divergence-free condition on the intermediate solution $\bar{\mathbf{u}}^*$ is enforced by a Helmholtz projection

  $$\bar{u}_i^{**} = \bar{u}_i^* - \partial_i \phi. \tag{65}$$

  Using (32) and the continuity equation one requires

  $$\partial_k \bar{u}_k^{**} = \partial_k \tilde{u}_k^{**} - \mathcal{U}_k \partial_k W = 0, \tag{66}$$

  and applying the divergence operator to (65) using (66) we obtain

  $$\partial_k \bar{u}_k^* = \partial_k \partial_k \phi. \tag{67}$$

- **(W3)** The final dewindowing is then performed as

  $$\bar{u}_i^{(n+1)} = \bar{u}_i^{***} = (1 - W)\mathcal{U}_i + W\bar{u}_i^{**}. \tag{68}$$

  The divergence error of the solution due to dewindowing at time $t^{(n+1)}$ is

  $$\partial_i \bar{u}_i^{(n+1)} = (\bar{u}_i^{**} - \mathcal{U}_i)\partial_i W. \tag{69}$$

  It is non-vanishing only in the windowing regions where, however, $(\bar{u}_i^{**} - \mathcal{U}_i)$ is small due to **(A2)**.

The Poisson equation (67) can be solved accurately by a Fourier transform only if $\bar{\mathbf{u}}^*$ is periodic on Γ. A non-periodic $\bar{\mathbf{u}}^*$ can be treated directly using a non-periodic Poisson solver of high order (e.g. [5]). Alternatively, $\bar{u}_k^*$ in Eq. (67) can be split into a periodic part and a non-periodic part. Formally assuming the additional dewindowing in **(W1)** given in Eq. (64), $\bar{u}_k^*$ can be written as

$$\bar{u}_k^* = (1 - W)\mathcal{U}_k + \tilde{u}_k^*, \tag{70}$$

$$= \underbrace{\mathcal{U}_k}_{\text{non-periodic}} \; \underbrace{-W\mathcal{U}_k + \tilde{u}_k^*}_{\text{periodic}}. \tag{71}$$

Since the base flow $\mathcal{U}_k$ satisfies continuity, $\partial_k \mathcal{U}_k = 0$, Eq. (67) simplifies to

$$\partial_k \bar{u}_k^* = \partial_k\left(\mathcal{U}_k - W\mathcal{U}_k + \tilde{u}_k^*\right) = \partial_k\left(-W\mathcal{U}_k + \tilde{u}_k^*\right) = \partial_k \partial_k \phi_p, \tag{72}$$

which is a periodic problem on Γ and can thus be solved in Fourier space with periodic boundary conditions. The new solution is then given as

$$\bar{u}_i^{**} = \bar{u}_i^* - \partial_i \phi_p \tag{73}$$

with ϕ_p denoting the periodic part of ϕ.

The pressure is obtained from

$$\Delta p = -\rho \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \tag{74}$$

with suitable boundary conditions (see e.g. [24]). If the pressure is not required on the entire domain, e.g., a windowing operation can be employed to restrict the solution to the physically meaningful part of the domain and to allow for an efficient (spectral) solution.

5. Application to 2D flows

Two numerical test cases involving the Navier–Stokes equations are considered hereafter. The first of these test cases is employed as a comparison of the fringe and windowing technique in terms of the damping characteristics of a single localised disturbance at the outflow boundary. The second test case examines the ability of both methods to return a spatially evolved flow back to the prescribed inflow condition. The combination of these two test cases can be viewed as a model for real applications of this type of artificial boundaries; e.g. in spatial boundary layer transition simulations the turbulent fluctuations at the outflow need to be damped and the laminar inflow boundary layer profile has to be reestablished accurately.

5.1. Single vortex

The first test case to be considered consists of a single vortex which is convected through the two-dimensional flow domain with unit velocity in the x_1 direction (plug flow). The computational domain Γ is $[0,4\pi]\times[0,2\pi]$ and the Reynolds number is set to infinity (inviscid flow). The vorticity distribution of the vortex is based on [23] and given as

$$\omega(r) = \frac{C}{R^2}\sqrt{\mathrm{e}^{-(r/R)^2}}\left(\frac{r^2}{R^2}-2\right), \tag{75}$$

where $r=\sqrt{(x_1-\pi)^2+(x_2-\pi)^2}$. The initial location is at $x_1 = x_2 = \pi$ and the parameters are set to $C = 1$ and $R = 1/2$ (see Fig. 9). The total circulation in the domain vanishes. The velocity field induced by the vortex is superimposed onto the uniform convection velocity $u_1 = 1$, $u_2 = 0$. The exact solution of the given problem is the downstream convection of the vortex without deformation. Since the velocity induced by the vortex decays rapidly with increasing radial distance, the exact solution far from the vortex core is the constant convection velocity $u_1 = 1$.

In the x_2 direction, periodic boundary conditions are imposed. For the inhomogeneous direction x_1, the boundary treatment with both the fringe and the windowing technique has been used to damp out the vortex as it leaves the physical domain and to prevent the vortex from reentering. An exponential window function similar to (21) has been used. The base flow is thus $\mathscr{U}_x = 1$ and $\mathscr{U}_y = 0$. An ideal boundary treatment would completely cancel out the vortex at the outflow without disturbing the inflow condition or reflecting back into Γ_I.

The domain is discretised by 128×64 grid points in the x_1 and x_2 directions, respectively. Corresponding simulations with higher resolution confirmed that the results are grid-independent. In order to quantify the global accuracy and especially the damping behaviour of the two artificial boundary techniques, the L_2 error in the physical subdomain $\Gamma_I = [0, \frac{2}{3}4\pi] \times [0, 2\pi]$ is calculated,

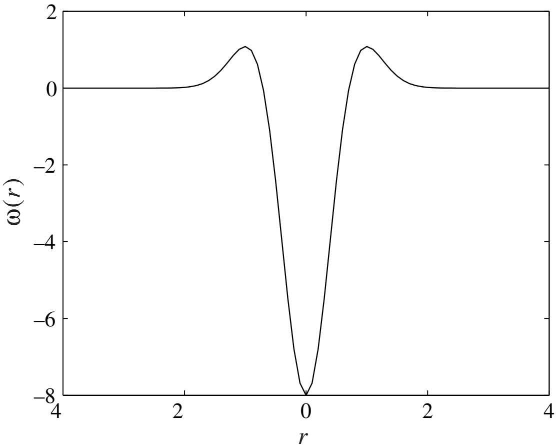

Fig. 9. Vorticity distribution of the single vortex defined in Eq. (75).

$$L_2(\mathbf{u}) = \left[\frac{1}{l_x l_y}\int_0^{l_x}\int_0^{l_y} |\mathbf{u}-\mathbf{u}_{\text{ex}}|^2 \,\mathrm{d}x\,\mathrm{d}y\right]^{1/2} \tag{76}$$

with **u** and $\mathbf{u}_{\text{ex}}$ denoting the computed and the exact solution, respectively. The dimensions of the integration area are $l_x = 8\pi/3$ and $l_y = 2\pi$. Similarly, the x-dependent error is defined as

$$L_{2,x}(\mathbf{u},x) = \left[\frac{1}{l_y}\int_0^{l_y} |\mathbf{u}-\mathbf{u}_{\text{ex}}|^2 \,\mathrm{d}y\right]^{1/2}. \tag{77}$$

A number of different fringe and windowing parameters were examined in the simulations A to K, see Table 1.

The temporal evolution of the L_2 error for a variation of the fringe strength (simulations A to D) is shown in Fig. 10. The initially low deviation between the computed and the exact solution increases as the vortex travels downstream through the domain Γ_I due to small discretisation errors. As soon as the vortex reaches the outflow of the physical domain ($x = \frac{8}{3}\pi \approx 8.4$, $t \approx 5$) the fringe forcing starts to damp the vortex towards zero. The following peak of the L_2 error ($t \approx 8$) expresses the upstream influence of this

Table 1
Different fringe and windowing parameters for test case 1 (single vortex)

Fringe	$\lambda_{\max}$	x_{start} (i)	Window	x_l (i)	x_r (i)
A	25	7.9 (80)	J	3.9 (40)	8.6 (88)
B	50	7.9 (80)	K, K$^+$	4.9 (50)	7.7 (78)
C	75	7.9 (80)			
D	100	7.9 (80)			
E	50	8.8 (90)			
F	50	9.8 (100)			
G	50	10.8 (110)	Fringe	% rise	% fall
H	50	7.9 (80)		3	3
I	34	7.9 (80)		3	3

Fringe parameters according to Eq. (6) with $x_{\text{end}} = 12.6$ (128); windowing see Eq. (21) with $x_L = 0$ (1) and $x_R = 12.6$ (128). Coordinates given in physical units and grid points (in parentheses).

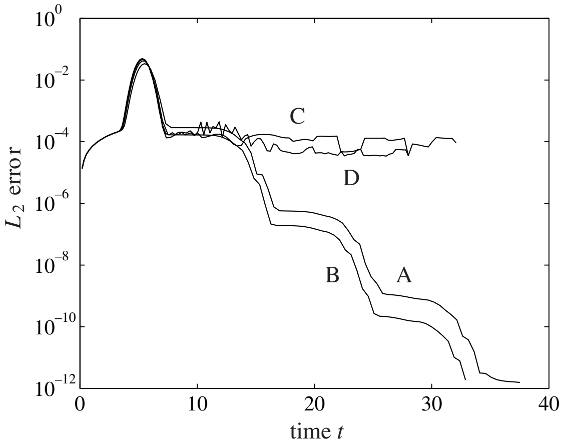

Fig. 10. L_2 error for test cases A to D: variation of the fringe amplitude $\lambda_{\max}$.

forcing. The stepwise decay of the L_2 error at intervals of $\Delta t \approx (8/3)\pi$ then shows the subsequent damping of the artefacts of the vortex that were able to penetrate through the fringe region and reenter the physical domain.

As can be seen from Fig. 10, the fringe strength does not play an important role in terms of the upstream influence of the fringe region, i.e. the error peak is similar for all four cases. On the other hand, the long-term damping behaviour is quite different. Case B is more strongly damped than case A as expected. An important feature is shown for cases C and D: Although the fringe amplitude is only increased moderately, the fringe forcing does not provide a weakening of the vortical disturbance at all; strong velocity fluctuations are generated within the fringe region. This behaviour could be prevented by a reduction of the time step according to the criteria presented in Eq. (8), but in the present implementation the time step is calculated based on the unmodified equation (1) alone. The inverse proportionality of the time step and the fringe strength $\lambda_{\max}$ is unlikely to allow an efficient solution procedure for high fringe amplitudes.

The influence of the starting point of the fringe region for a fixed fringe amplitude $\lambda_{\max}$ is shown in Fig. 11 corresponding to simulations B and E to G. The results confirm the observations made in Fig. 10: The earlier the fringe forcing starts, the more distinctively the initial error peak indicates a strong upstream influence. On the other hand, beginning the fringe forcing further downstream reduces this influence since it is more and more confined to the fringe region and does not appear in the L_2 error. However, a shorter fringe region results in a weaker damping of the error and the intervals between subsequent reductions of the error increase corresponding to x_{start}.

As already reported in [20], the shape of the fringe function is not very important for the overall damping behaviour. Fig. 12 shows simulations H and I in which the fringe function is basically rectangular.

Due to the higher total strength of the fringe (area below the fringe function) in simulation H, both the maximum error and the damping is increased. Simulation I reduced $\lambda_{\max}$ to match the fringe strength to the one of simulation B, but the initial error peak remains slightly stronger than in case B.

The results of the windowing technique are shown in Figs. 13 and 14. The inflow windowing region Γ_I^A is part of the physical domain Γ_I in order to be able to prescribe an exact inflow condition. The outflow windowing region is located at the right-hand end of the computational domain mainly within Γ_A and is therefore not part of the physically relevant solution (refer to Fig. 4). The shape of the window function (see Fig. 3 and Eq. (21)) justifies this procedure as the window function reaches values close to unity within a fraction of the total length $x_l - x_L$ or $x_R - x_r$.

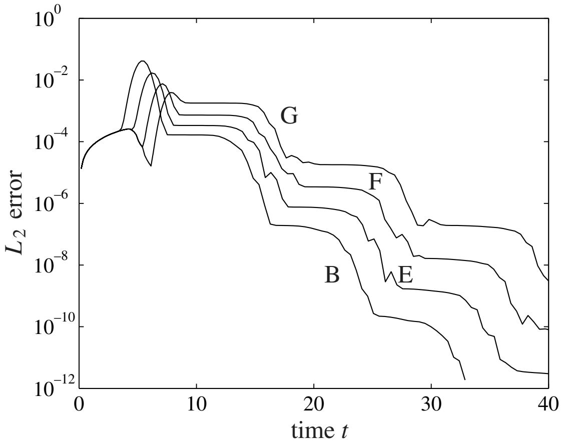

Fig. 11. L_2 error for test cases B, E to G: variation of the fringe region start x_{start}.

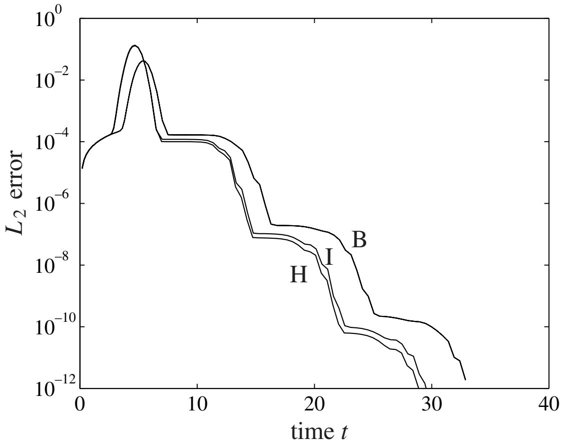

Fig. 12. L_2 error for test cases H, I and B: variation of the fringe shape.

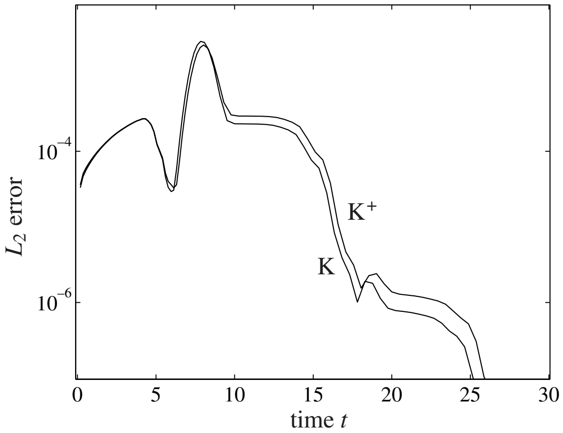

Fig. 13. L_2 error of the windowing test cases K and K$^+$: influence of the additional dewindowing step.

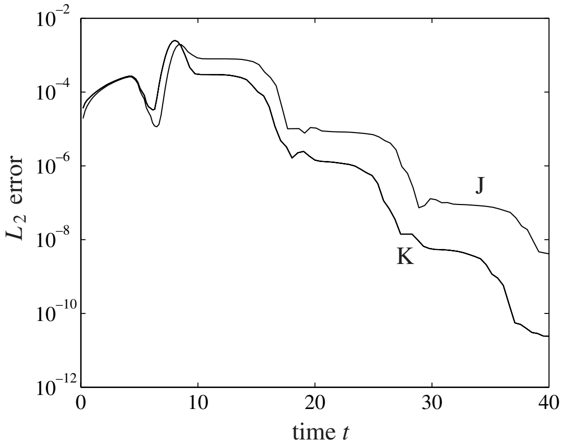

Fig. 14. L_2 error of test cases J and K: variation of the extent of the window region.

Fig. 13 shows the comparison of the two different simulations K and K$^+$ using the same window function. Run K does include the intermediate dewindowing step (**W1**), whereas run K$^+$ does not. Obviously, the accuracy of the results can be increased slightly by this step. Therefore, all subsequent windowing simulations do use this additional dewindowing step (see Section 4.2).

The results of the simulations for the windowing technique with different extents of the window function are shown in Fig. 14 for two symmetric windows. For $t > 5$, both cases show similar behaviour as for the fringe cases F and G, first a decrease of the error, indicating that the vortex has left the physical domain but has not yet been absorbed in the window. But as soon as the vortex is being damped, the L_2 error increases. The subsequent damping of the windowing scheme is mainly dependent on the length of the windowing region, i.e. the gradient of the window function; the decay of the error for case K is thus slightly faster than for case J.

How the error evolves during the passage of the vortex through the right-hand boundary is illustrated in Fig. 15 for both methods. In principle, both plots show a similar evolution. The vortex is convected through the physical domain as indicated by a slightly increased error due to the numerical truncation. The absorption zone is reached at $t \approx 6$ (fringe) and $t \approx 8$ (windowing). The passage causes an immediate upstream spike of the error, which is similar in shape and strength for both techniques. Synchronously with the beginning of the forcing, a second region of increased error is generated at the left (inflow) boundary which is subsequently convected downstream with the uniform convection speed $\mathscr{U}_x = 1$. The main difference between the two methods demonstrated by this plot is the inherent feature of the windowing technique to provide the correct (prescribed) solution at the inlet as well as at the outlet of the computational domain. Moreover, it becomes evident that the main forcing of the window technique is confined to a region closer to the outflow boundary. Therefore, the influence of the forcing onto the physical domain Γ_I is smaller.

A closer comparison of the most successful parameter sets for the fringe and windowing approach is given in Fig. 16. Both schemes are comparable with respect to the long-term damping of the vortex, although the maximum damping is reached by the fringe method. On the other hand, the immediate upstream influence caused by the vortex entering the damping zone is higher for the fringe method.

To conclude: Both techniques allow a comparable damping of a single localised disturbance. Although the fringe method can reach a higher damping rate with optimally chosen parameters, its upstream influence is higher than it is for the windowing approach. Moreover, imprudent choice of fringe parameters can lead to an unstable scheme unless further restrictions on the time step are made. In this respect, the windowing technique is more robust.

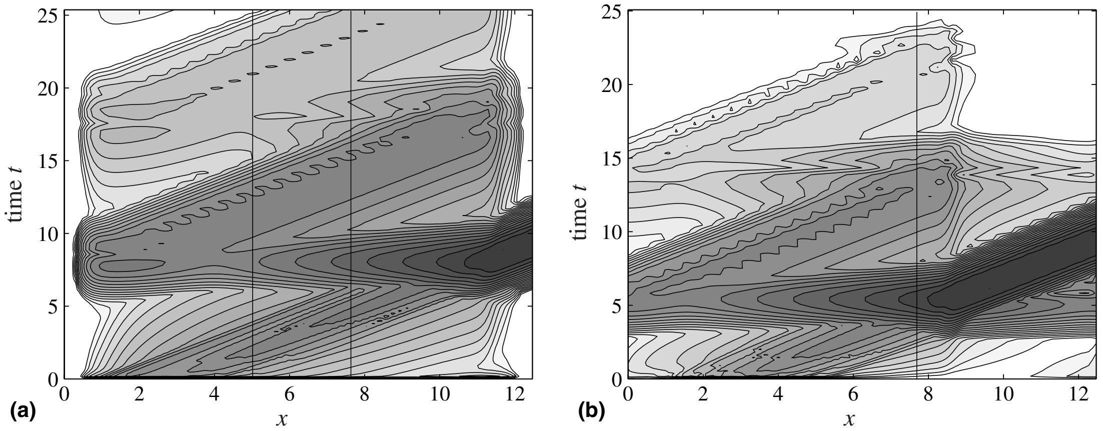

Fig. 15. Contours of the local error $L_{2,x}$ with (a) windowing (case J) and (b) fringe technique (case B) in a (x, t) plane. 20 contour lines of $\log_{10}(L_{2,x})$ ranging from −15 (light) to 1.6 (dark).

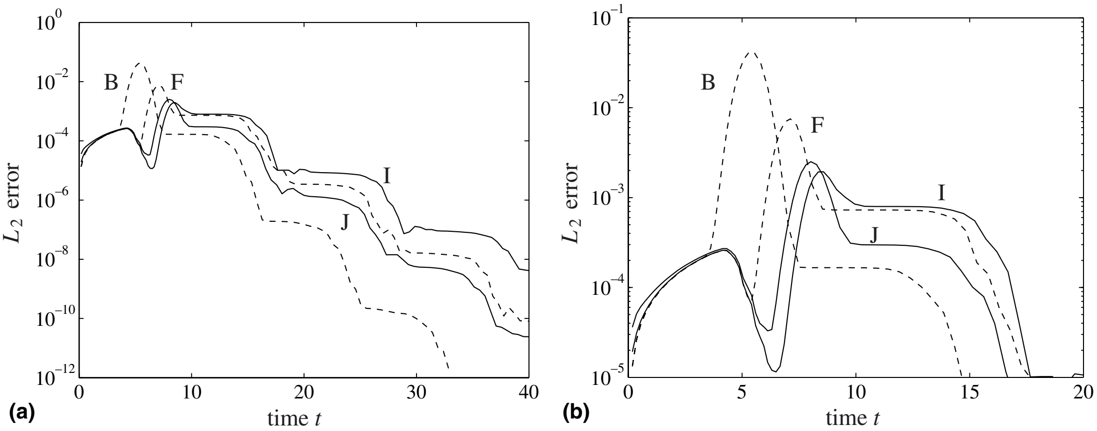

Fig. 16. Comparison of the ---- fringe (cases B and F) and — windowing technique (cases I and J) for the single vortex test case. (b) Close-up of (a).

5.2. Planar jet

The second test case consists of a laminar planar jet with no superimposed disturbances. Periodic boundary conditions are imposed in the cross-stream direction. The flow configuration is stationary and spatially evolving in the streamwise direction, providing a test case for the ability of both the fringe and windowing technique to return a spatially evolved flow back to its prescribed inflow state. The inflow and outflow conditions are based on the analytical solution presented in [25] for a planar jet flow leaving an infinitesimally small orifice. Starting from this analytical solution, a reference simulation was run on a domain $[0,64] \times [0,8]$ with 256×32 grid points, in which the streamwise direction was made periodic using the fringe technique. The base flow used in this test case was then extracted from the reference simulation. Fig. 17 shows the streamwise and cross-stream velocity contours of the base flow.

In contrast to test case 1 presented in the previous section, this base flow is non-periodic in the streamwise direction. Therefore, an adapted periodic base flow has to be constructed for the fringe technique and

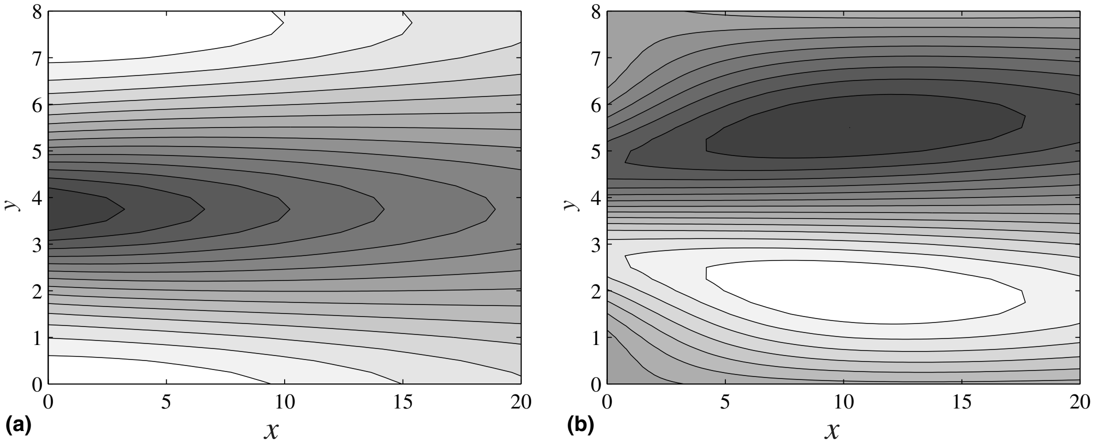

Fig. 17. Contour lines of the planar jet used as base flow for test case 2. *Left:* Streamwise velocity u, 15 contour levels between −0.3 and 1.3 *Right:* Cross-stream velocity v, 15 contour levels between −0.03 and 0.03, high velocity corresponds to dark colour.

the periodic windowing technique (case 4 in Section 3). A blending region of the form (5) is added to the streamwise velocity component of the base flow. The cross-stream component in the blending region is calculated using continuity.

The dimension of the flow domain is [0,32] × [0,8] and, similarly to test case 1, the L_2 error is computed on the subdomain [0,20] × [0,8] of the computational domain. The spatial directions are discretised using 128 × 32 grid points. The Reynolds number is set to 30 which results in an appreciably broad downstream jet spreading.

Several simulations have been conducted as listed in Table 2. Simulations A to Q have been performed using a periodic base flow as just described. For simulations G to I and O to Q (windowing technique), the blending region was positioned completely in Γ_A where $W(x)$ was significantly larger than zero. In a first group of simulations (cases A to I), all velocity components were blended using Eq. (5) without correction of the violated continuity equation. Conversely, in simulations J to Q the cross-stream velocity component v was adapted according to continuity.

Simulations R to V applied the windowing method without a periodic base flow exploiting the feature of the windowing technique that the transformed Navier–Stokes equation (38) can be written in a way that they do not contain spatial derivatives of non-windowed quantities and therefore do not rely on a periodic base flow.

Again, the influence of the additional dewindowing step **(W1)** has been examined. Based on the results given in Table 3 all subsequent runs do in fact include this step.

5.2.1. Blending without divergence correction: cases A to I

The runs A to I present the results of the fringe and windowing technique with the base flow made periodic using a blending. All components of the base flow $\mathcal{U}$ were blended similar to Eq. (5), thus in general $\partial_i \mathcal{U}_i \neq 0$.

Table 2
Different fringe and windowing parameters for test case 2 (planar jet)

	Fringe method				Windowing method		
	$\lambda_{\max}$	x_{start} (i)	L_2 error		x_l (i)	x_r (i)	L_2 error
A	30	20 (80)	0.010	G	10 (40)	22 (88)	7.64×10^{-4}
B	30	22.5 (90)	0.017	H	12.5 (50)	19.5 (78)	3.45×10^{-4}
C	30	25 (100)	0.031	I	15 (60)	17 (68)	4.51×10^{-4}
D	10	20 (80)	0.010				
E	1	20 (80)	0.018				
F	0.1	20 (80)	0.164				
J	30	20 (80)	1.47×10^{-4}	O	10 (40)	22 (88)	4.18×10^{-4}
K	30	22.5 (90)	1.82×10^{-4}	P	12.5 (50)	19.5 (78)	1.12×10^{-4}
L	30	25 (100)	2.57×10^{-4}	Q	15 (60)	17 (68)	2.27×10^{-4}
M	10	20 (80)	3.55×10^{-4}				
N	1	20 (80)	6.48×10^{-4}				
				R	5 (20)	27 (108)	3.04×10^{-4}
				S	7.5 (30)	24.5 (98)	9.79×10^{-5}
				T	10 (40)	22 (88)	8.14×10^{-5}
				U	12.5 (50)	19.5 (78)	6.04×10^{-5}
				V	15 (60)	17 (68)	3.66×10^{-5}

Fringe parameters according to Eq. (6) with $x_{\text{end}} = 32$ (128); windowing see Eq. (21) with $x_L = 0$ (1) and $x_R = 32$ (128).
Coordinates given in physical units and grid points (in parentheses).

Table 3
Influence of additional dewindowing step (**W1**)

Option	L_2 error	
Additional dewindowing	Periodic	Non-periodic
No	1.37×10^{-4}	6.30×10^{-5}
Yes	1.12×10^{-4}	6.04×10^{-5}

Results are given for both periodic (case P) and non-periodic base flow (case U).

The stationary error after $t = 100$ with variation of the starting point of the fringe forcing is shown in Fig. 18. Consistent with the test case presented in Section 5.1, increasing x_{start} results in an overall weaker forcing towards the desired solution. On the other hand, the immediate upstream influence into the domain Γ_I can be reduced. However, as Table 2 and Fig. 18 clearly show, the total L_2 error does increase with a shorter fringe region.

Fig. 19 shows the evolution of the maximum streamwise velocity component ($y = 4$) over the computational domain. At $x = 0$ the difference between the prescribed and the computed solution is more than 2%. For velocity profiles closer to the cross-stream boundary (not shown), a similar behaviour is observed: u is overestimated within the fringe region leading again to an error of roughly 2%. The streamwise velocity does not follow the base flow close to and within the fringe region; u increases earlier (upstream influence) than the prescribed solution, and does not reach the maximum value of the blended base flow. The main reason for this behaviour and the resulting large L_2 errors lies in the base flow which is not satisfying continuity: The flow in the fringe region is forced towards a non-incompressible solution. Since continuity is necessarily enforced in the flow solution (via the projection step in the numerical algorithm), errors are introduced which reduce the accuracy in the whole domain.

Conversely, the windowing technique is not very sensitive in that respect, since the prescribed solution is enforced in the dewindowing step and the possibly violated continuity equation is not corrected thereafter. The stationary error for the windowing simulations (cases G to I) is displayed in Fig. 20. The global L_2 error decreases with larger window size, although an increased upstream influence can be observed in Fig. 20. The global L_2 error of the windowing approach is, for any set of reasonable parameters, at least two decades smaller than for the fringe technique.

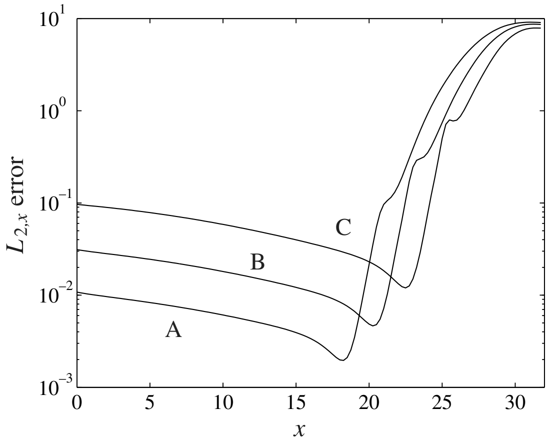

Fig. 18. $L_{2,x}$ error for test cases A to C: variation of the fringe start x_{start}.

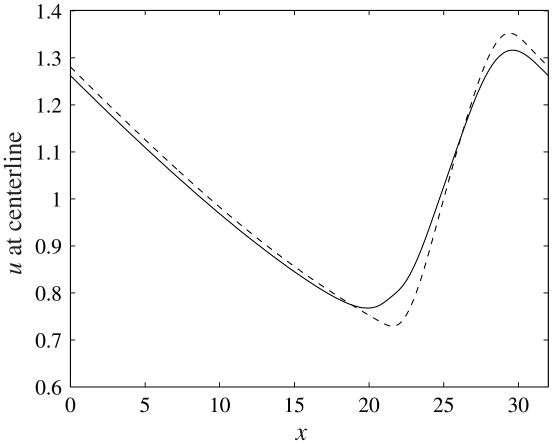

Fig. 19. Evolution of the maximum streamwise velocity u over the computational domain for the fringe method (test case A, $x_{\text{start}} = 20$). ---- prescribed solution, — numerical solution.

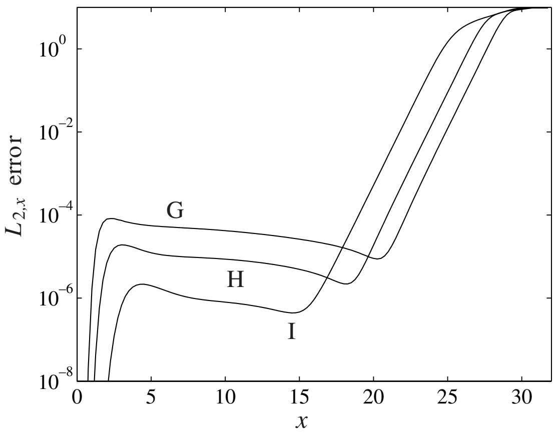

Fig. 20. $L_{2,x}$ error for test cases G to I: variation of the extent of the windowing region.

5.2.2. Blending of mean flow satisfying continuity: cases J to Q

In contrast to the cases just described, simulations J to Q are performed with a base flow satisfying continuity. Note that the correction is only needed in the blending region (i.e. where $\mathrm{d}S(x)/\mathrm{d}x \neq 0$ in Eq. (5)), the continuity equation is satisfied in the rest of the domain by construction of the base flow (converged solution).

Analogously to Fig. 18, Fig. 21 compares the error of the fringe method for different starting points of the forcing. As is to be expected, the global error level is much smaller than for the cases A to I. Additionally, the total error increases with a later start of the forcing, but compared to cases K to M it seems to be less sensitive. The upstream influence of the fringe forcing also increases with an earlier start of the forcing. The influence of the fringe strength can be seen in Fig. 22. Stronger forcing causes higher upstream influence but overall smaller errors.

For the windowing method, the comparison between Figs. 20 and 23 does not show major differences, neither qualitatively nor quantitatively. The conclusion is that for the windowing technique the knowledge

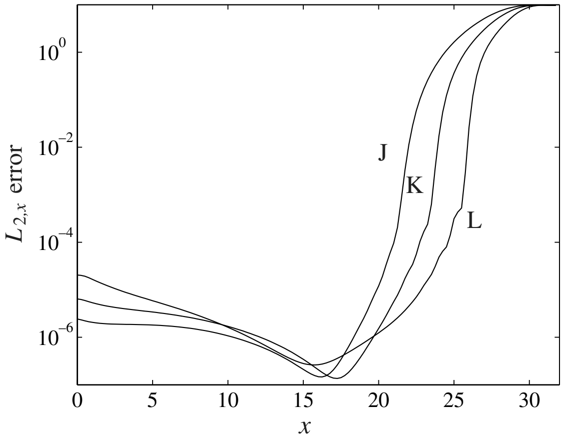

Fig. 21. $L_{2,x}$ error for cases J to L: variation of the fringe start x_{start}.

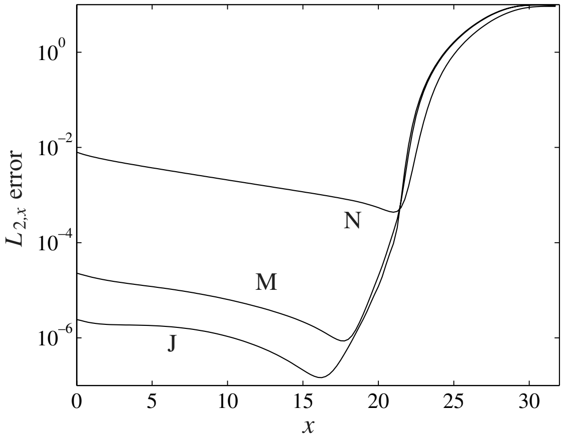

Fig. 22. $L_{2,x}$ error for cases J, M and N: variation of the fringe strength $\lambda_{\max}$.

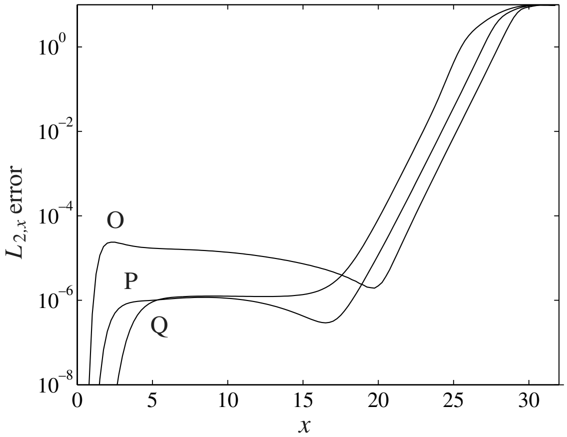

Fig. 23. $L_{2,x}$ error for cases O to Q: variation of the extent of the windowing region.

of a physically correct solution within the windowing regions is not as important as for the fringe method. The reason for that is given by the fact that the Navier–Stokes equations are only solved on the windowed quantities and the boundary condition is added to the windowed solution (see Eq. (32)). Therefore, the influence of a physically incorrect outflow solution is minimal as long as the conditions in Γ_I^I present a suitable and correct inflow.

In Fig. 24(a) drastic difference to Fig. 19 can be stated. With a physical solution in the fringe region, the fringe method is obviously able to force towards a prescribed state, although a small spatial delay can be observed. The error of the maximum streamwise velocity at the inflow position has decreased from 2% in case K to below 10^{-3} for simulation T. Both the upstream influence and the error at the inflow are reduced substantially. The error for the windowing technique is basically constant compared to case H (about 10^{-3}).

5.2.3. Non-periodic base flow: cases R to V

Cases R to V are different from all the other cases in that the base flow $\mathscr{U}_i$ is not artificially made periodic. As derived in Section 4.2, changes in the numerical algorithm allow directly the use of a non-periodic

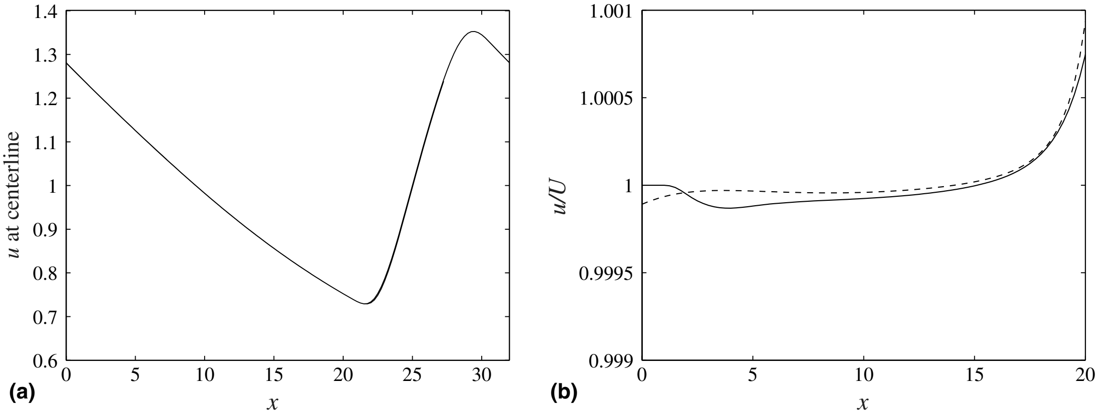

Fig. 24. Evolution of the maximum streamwise velocity u over the computational domain. *Left:* ---- fringe technique compared to the — blended base flow $\mathscr{U}$. *Right:* ratio $u/\mathscr{U}$ for ---- fringe and — windowing approach. Test cases K and P.

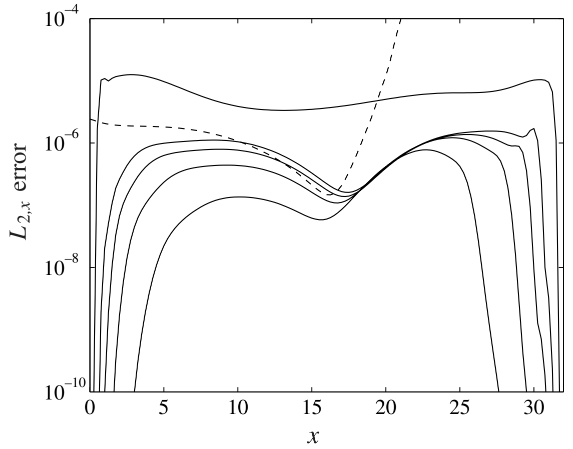

Fig. 25. — $L_{2,x}$ error for cases R to V: variation of the extent of the windowing region in the non-periodic case (cases R to V with decreasing error). For comparison: ---- error of case A (fringe technique).

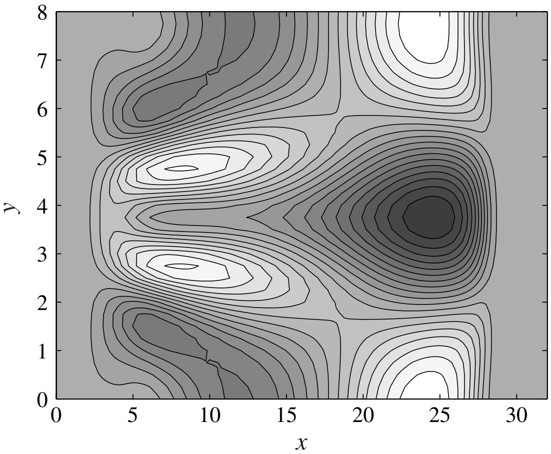

Fig. 26. Deviation from the mean flow $u - \mathscr{U}$ for case U. Plotted 20 contours ranging from -1.7×10^{-4} (light) to 2.2×10^{-4} (dark).

base flow without losing the accuracy and efficiency of Fourier methods. Therefore, no systematic comparison to the fringe method is possible for these test cases as for the fringe technique always a periodic base flow is required.

Fig. 25 clearly shows the advantage of the non-periodic windowing method. Basically the whole domain Γ can be used as a physical solution. Moreover, the L_2 error is smaller for most of the window parameters. It can therefore be concluded that the overall efficiency of this method is superior to that of the fringe technique and to the periodic windowing method.

Fig. 26 shows the deviation of the computed solution from the mean flow $u - \mathscr{U}$. The error is distributed evenly around zero. Moreover, it is clearly visible that the windowing method upholds the exact boundary conditions near the inflow and outflow domain boundaries.

6. Conclusions

The windowing technique presented in this report, adapted from [14] and refined by introducing a regularised dewindowing procedure, proved successful in solving non-periodic problems governed by the Navier–Stokes equations while employing periodic Fourier discretisation. Analytical predictions on the convergence rate of the numerical solution can be made, which are confirmed by our numerical results. The convergence rate depends fundamentally on the choice of the windowing function. Properly designing this function allows to recover spectral accuracy [13]. An adaptation of the present algorithm to compressible flows is possible. The exact satisfaction of inflow conditions is ensured due to the dewindowing procedure even for base flows which do not accurately satisfy continuity. For this type of (non-physical) base flow the fringe method produces inferior results.

The accuracy of the windowing and the fringe method (see [20]) was compared for a number of parameters and three test cases. They cover essential requirements encountered in real applications, e.g. spatial laminar-turbulent transition and turbulence simulations. Both methods were successful in providing an accurate inflow condition and an artificial outflow condition with limited upstream influence. The global errors of the windowing method were found to be at least as low as the errors of the fringe technique.

It was found that the performance of the fringe method strongly depends on the choice of the parameters: fringe strength, blending, start and length of the fringe region. While excellent results can be obtained for ideal parameters, the imprudent choice of parameters can even lead to numerical problems due to the fringe forcing. The reduction of the time step in such cases is undesirable as it decreases the efficiency of the computation significantly. The windowing method, on the other hand, does contains a small number of tuning parameters and their influence is limited.

Implementing the fringe method into an existing simulation code does not pose any problems. For the windowing method, additional terms involving derivatives have to be included in the differential equation, which is more complicated. The cost of the additional calculations for the fringe forcing is negligible, and that for the evaluation of the windowed quantities and fluxes is small (increase of CPU time due to windowing operations around 5%). For both methods, the fraction of the computational domain needed for the treatment of the artificial boundary conditions is similar. However, for spatially evolving, non-periodic base flows the fringe method relies on a blending to provide a periodic base flow. The windowing method, on the other hand, can be applied directly using a non-periodic base flow and does not necessarily need an extension of the domain by a blending region.

The influence of the fringe region on the accuracy is difficult to estimate. For the windowing technique, simple estimates can be used to assess the impact of the boundary treatment on the accuracy. These relations show that the accuracy can be increased with finer resolution of the window function.

The windowing method provides an attractive alternative way to perform accurate simulations in non-periodic geometries using periodic discretisation schemes. Since it poses no restriction on the type of inflow conditions, its use for transitional and turbulent flows is well possible.

Acknowledgements

This work was supported by the Swiss National Science Foundation (SNF) and the Swiss National Supercomputing Centre (CSCS).

References

[1] J.-P. Bérenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (2) (1994).

[2] F.P. Bertolotti, T. Herbert, P.R. Spalart, Linear and nonlinear stability of the Blasius boundary layer, J. Fluid Mech. 242 (1992) 441–474.

[3] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover, Mineola, USA, 2000.

[4] L. Brandt, P. Schlatter, D.S. Henningson, Boundary-layer transition under free-stream turbulence, J. Fluid Mech. 517 (2004) 167–198.

[5] E. Braverman, M. Israeli, A. Averbuch, L. Vozovoi, A fast 3D Poisson solver of arbitrary order accuracy, J. Comput. Phys. 144 (1998) 109–136.

[6] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.

[7] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer, Berlin, Germany, 1988.

[8] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[9] T. Colonius, Modeling artificial boundary conditions for compressible flow, Annu. Rev. Fluid Mech. 36 (2004) 315–345.

[10] T. Colonius, H. Ran, A super-grid scale model for simulating compressible flow on unbounded domains, J. Comput. Phys. 182 (2002) 191–212.

[11] W. E, C.-W. Shu, A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow, J. Comput. Phys. 110 (1994) 39–46.

[12] D. Givoli, Non-reflecting boundary conditions, J. Comput. Phys. 94 (1991) 1–29.

[13] Y. Guo, A new Fourier spectral method for the direct numerical simulation of three-dimensional, spatially growing compressible boundary layer transition, Technical Report IB 221-93A16, DLR Göttingen, Germany, 1993.

[14] Y. Guo, N.A. Adams, L. Kleiser, Direct numerical simulation of transition in a spatially growing compressible boundary layer using a new Fourier method, in: P. Voke, L. Kleiser, J.-P. Chollet (Eds.), Direct and Large-Eddy Simulation I, Kluwer Academic Publishers, The Netherlands, 1994, pp. 249–259.

[15] T. Hagstrom, I. Nazarov, Perfectly matched layers and radiation boundary conditions for shear flow calculations, AIAA Paper 2003-3298, 2003.

[16] H.F. Harris, On the use of windows for harmonic analysis with the discrete Fourier transform, Proc. IEEE 66 (1) (1978) 51–83.

[17] M. Israeli, S.A. Orszag, Approximation of radiation boundary conditions, J. Comput. Phys. 41 (1981) 115–135.

[18] M. Israeli, L. Vozovoi, A. Averbuch, Spectral multidomain technique with local Fourier basis, J. Sci. Comput. 8 (2) (1993) 135–149.

[19] A. Lundbladh, S. Berlin, M. Skote, C. Hildings, J. Choi, J. Kim, D.S. Henningson, An efficient spectral method for simulation of incompressible flow over a flat plate, Technical Report TRITA-MEK 1999:11, KTH Stockholm, Sweden, 1999.

[20] J. Nordström, N. Nordin, D.S. Henningson, The fringe region technique and the Fourier method used in the direct numerical simulation of spatially evolving viscous flows, SIAM J. Sci. Comput. 20 (4) (1999) 1365–1393.

[21] R.K. Otnes, L.D. Enochson, Applied Time Series Analysis, Wiley–Interscience, New York, USA, 1978.

[22] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.

[23] T.J. Poinsot, S.K. Lele, Boundary conditions for direct simulations of compressible viscous flows, J. Comput. Phys. 101 (1992) 104–129.

[24] R. Rannacher, On Chorin's projection method for the incompressible Navier–Stokes equations, in: Navier–Stokes Equations II – Theory and Numerical Methods, Lecture Notes in Mathematics, vol. 1530, Springer, Berlin, Germany, 1992, pp. 167–183.

[25] H. Schlichting, Boundary-Layer Theory, seventh ed., McGraw-Hill, New York, USA, 1987.

[26] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[27] P.R. Spalart, Direct numerical study of leading edge contamination, in: Fluid Dynamics of Three-Dimensional Turbulent Shear Flows and Transition, AGARD-CP-438, AGARD, Neuilly-sur-Seine, France, 1988, pp. 5.1–5.13.