

A sliding characteristic interface condition for direct numerical simulations.

Roderick Johnstone^{a,*}, Liwei Chen^a, Richard Sandberg^a

^a*Aerodynamics and Flight Mechanics Research Group, Faculty on Engineering and the Environment, University of Southampton, Southampton SO17 1BJ, UK*

Abstract

A characteristic interface condition serves as the basis for a novel sliding grid method, with a view to solving the compressible Navier-Stokes equations on block-structured grids that are delimited by boundary conditions in motion relative to each other. This requires that the convective and source terms of the equations in characteristic form be transformed to the reference frame of the neighbouring block, and interpolated. The method facilitates accurate interpolation at the interface, because the characteristic interface condition requires only a single layer of halo nodes. When a homogeneous direction is present, only 1-D interpolation is required, and schemes that might otherwise be too costly become affordable. The treatment also enjoys the same advantages as fixed characteristic interfaces do in relation to tolerance of grid discontinuities at block interfaces. The implementation and parallelisation of this method in a simulation code is described, and accuracy and performance demonstrated on a selection of test cases.

Keywords: Compressible Navier-Stokes equations, Characteristic interface conditions, Sliding grids, Parallel computing

1. Introduction

Sliding grids are a natural solution to some of the problems of simulating unsteady turbomachinery flows [7, 21], as well as simulations of rotorcraft [25], moving control surfaces [6], and stirred-tank reactors [38]. This approach makes it possible to discretise the flow around each body in the

*Corresponding author. Tel.: +44 (0)23 8059 3394; fax: +44 (0)23 8059 3058.

reference frame in which the body is stationary. Then it becomes possible to make straightforward use of block-structured grids, and wavenumber-optimised finite-difference schemes (eg. [29, 13]) that are most naturally implemented on such grids, the advantages of which are particularly compelling in the direct numerical simulation of turbulence [14].

It is a fundamental property of turbulent flow that a wide range of length scales is represented, and this is what makes its direct numerical simulation (DNS) computationally demanding. Computational economy therefore dictates the use of methods suitable for marginal resolutions, i.e. high-order methods. It may be worthwhile, however, to sacrifice some formal accuracy at low wavenumbers (which are in any case well discretised) in order to improve that for the marginally-resolved high wavenumbers. In particular, the latter will suffer from dispersion error [29]; that is, the group velocity of the short waves as represented by the finite difference equations differs from that associated with the continuous equations they discretise. This issue is not important in the context of time-independent aerodynamic calculations, but becomes so when time accuracy is required [27], as is the case in DNS.

The dispersion error of a central finite difference scheme is (see [1]) the difference of the effective wavenumber k^* :

$$k^* = \frac{2}{\Delta x} \sum_{j=1}^N a_j \sin(jk\Delta x) \quad (1)$$

and the exact wavenumber k , for a $2N + 1$ -point scheme with coefficients a_j . The relationship between k and k^* is shown in figure 1. It is apparent that it improves rather slowly with the formal order of the scheme. Moreover, optimising a central scheme for dispersion error degrades its order [1]. These factors exert upward pressure on the size of the computational stencil. For this reason, compact schemes are preferred, even though their implementation in parallel codes is much less straightforward [13].

It is quite rare, however, for such finite difference schemes to find application in the context of a sliding mesh simulation. Steger and Benek [24] argue that a single point of overlap between two adjoining grids makes for a method that is as much an overset grid approach as a patched or sliding grid. Indeed, overlapping (or halo) nodes hinder grid generation even for fixed multiblock grids with conforming grid lines, as derivatives of spatial coordinates with respect to computational coordinates must still be continuous in the halo [12]. If this problem is addressed by implementing an overset method, interpola-

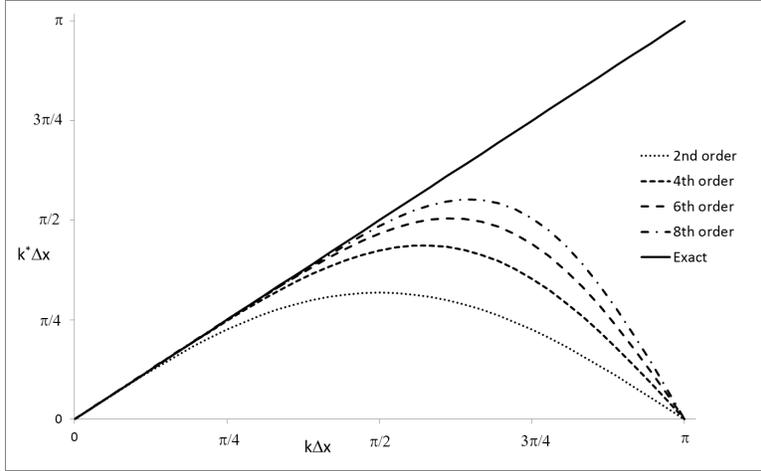


Figure 1: Effective wavenumber (k^*) vs. exact (k) for standard central difference schemes.

tion will be required at every halo node, and this becomes progressively less attractive as the size of the differentiation stencil (and with it the number of halo nodes) increases. The interpolation stencil may also need to be expanded to match the order of the method; ideally, if wavenumber-optimised finite difference schemes are used, it should be in combination with similarly constructed interpolation schemes (e.g. [28]), and these are computationally expensive (interpolation coefficients are obtained by solving a system of equations).

Perhaps partly for this reason, direct numerical simulations of turbulent flow have not made use of sliding grids, with the exception of the DNS of Rai [21], who simulated rotor-stator interaction using high-order upwind differencing. A sliding interface [20] was implemented using an iterative implicit scheme that imposes Dirichlet boundary conditions at the interface to decouple the calculations in the blocks on either side; the boundary values were corrected iteratively using cubic spline interpolation. Halo nodes also require interpolation. The need for iteration would appear to make this approach best suited to implicit time integration, which is typically used only for diffusive transport in direct numerical simulation of turbulence. Using this approach, it was possible to reproduce, for the most part, measured pressure coefficient and Stanton number distributions on the downstream blade. Rai noted that Karman vortices in the wake as it met the leading edge were

clearly visible; in this case, the turbulent wake contained significant coherent energy.

Rotor-stator interaction is a problem that lends itself to DNS, owing to the low Reynolds number, the need to simulate transition, and the presence of coherent structures; the use of sliding grids to discretise the flow is also very attractive. This is the application area that will be considered in the current paper. In earlier direct simulations of similar flows, Wu and Durbin [35] performed DNS of flow over a turbine blade subject to disturbance by turbulent wakes translating in the pitchwise direction. These wakes were introduced at the inflow boundary, rather than by simulating the hypothetical cylinders generating them. The wake turbulence was produced using a precursor simulation of a temporally decaying plane wake [36], believed to be a sufficiently good approximation to a cylinder wake sufficiently far downstream for the latter to have lost its shedding characteristics. The wakes were found to give rise to long, streamwise-aligned structures (in spite of the apparent similarity, not Görtler vortices) on the pressure side of the blade, although the boundary layer remained laminar.

Michelassi et al. [16] examined essentially the same flow using LES and DNS, finding that the incoming turbulent wakes intermittently suppressed a laminar separation on the suction side of the blade, near the trailing edge. The wakes did not appear to have any effect on the strongly accelerated flow over the upstream half of the suction side.

Cui et al. [3] also investigated this geometry, again using the wake precursor simulation of Wu et al. [36], and considering high levels of free stream turbulence (FST) in addition to that of the wake. It was found that the effect of the FST was to reduce the disturbance of the boundary layer by the wake, but that it did not replicate the effect of the wakes in respect of the long vortical structures observed by Wu and Durbin.

Tyagi and Acharya [31] used large eddy simulation to simulate rotor-stator interaction; the solid surfaces were represented using moving immersed boundaries. This study was intended, however, as a demonstration of the capabilities of their method, and the authors warn that it may not be representative of the true physical problem (for instance, that the spanwise domain width was relatively small at 0.1 chord lengths). They do argue, however, that this approach is a viable alternative to a sliding mesh.

Rotor-stator interaction has been studied much more extensively using RANS methods [4], notwithstanding the difficulty of adequately modelling intermittent bypass transition (caused by pitchwise-translating wakes) in this

way. For instance, Gagnon et al. [7] investigated rotor-stator interaction in an axial hydraulic turbine using unsteady RANS, and carried out spatial discretisation using second-order upwind differencing. The temporal discretisation was also second-order. A sliding mesh was used, and tested by simulation of vortex-shedding flow generated by a square cylinder. The sliding mesh was found to cause additional numerical diffusion, and grid refinement was required. This was in spite of the use of a turbulence model ($k-\epsilon$) that, as noted by the authors, was excessively diffusive for this flow even in the absence of a sliding mesh. It was, however, possible to obtain good agreement with experimental results of flow through an axial turbine.

We consider that possible obstacles to the use of high-order methods with sliding grids in the context of DNS include

- 1) Sliding grids are often dissipative (see [6], [7]); accurate interpolation would be required for DNS.
- 2) The use of large differentiation stencils would usually require deep haloes at sliding interfaces, which make the implementation of accurate interpolation methods difficult and computationally costly.
- 3) In the event that the grid is curvilinear, it may be difficult to maintain continuity of spatial coordinates with respect to computational coordinates [12] across a block or sliding interface, and this difficulty increases with the size of the halo.
- 4) DNS of turbulent flow is computationally demanding. Sliding interfaces complicate parallelisation, as efficient communication patterns must vary dynamically; excessively large haloes would also impede parallel performance.

In the current paper, we propose a new sliding grid technique, based on characteristic interface conditions, which addresses some of these drawbacks.

2. Method

The approach described here is based on the characteristic interface condition of Kim and Lee [12], which permits grid lines to change direction abruptly at interfaces, such that derivatives of spatial coordinates with respect to computational ones may be discontinuous. This greatly simplifies

grid generation for high-order differentiation, but requires the specification of boundary conditions on all faces of each block. Communication between adjacent blocks then takes place through these boundary conditions, but not through the differentiation stencil, which is one-sided at the interfaces. The boundary conditions at the interfaces are expressed in terms of the Navier-Stokes equations in characteristic form. In their conservative formulation these may be written:

$$\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \frac{\partial(\hat{\mathbf{E}}_i - \hat{\mathbf{E}}_v)}{\partial \xi} + \frac{\partial(\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)}{\partial \eta} + \frac{\partial(\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)}{\partial \zeta} = 0. \quad (2)$$

The caret denotes transformation to computational (ξ, η, ζ) space, \mathbf{E}_i , \mathbf{F}_i , \mathbf{G}_i are inviscid fluxes, \mathbf{E}_v , \mathbf{F}_v , and \mathbf{G}_v viscous.

$$\begin{aligned} \mathbf{Q} &= (\rho, \rho u, \rho v, \rho w, \rho e_t)^T \\ \mathbf{E}_i &= (\rho u, \rho u^2 + p, \rho v u, \rho w u, (\rho e_t + p)u)^T \\ \mathbf{F}_i &= (\rho v, \rho u v, \rho v^2 + p, \rho w v, (\rho e_t + p)v)^T \\ \mathbf{G}_i &= (\rho w, \rho u w, \rho v w, \rho w^2 + p, (\rho e_t + p)w)^T \end{aligned}$$

where e_t is the specific total energy, $\rho e_t = p/(\gamma - 1) + \rho(u^2 + v^2 + w^2)/2$, and

$$\begin{aligned} \hat{\mathbf{Q}} &= \mathbf{Q}/J \\ \hat{\mathbf{E}}_i &= (\xi_x \mathbf{E}_i + \xi_y \mathbf{F}_i + \xi_z \mathbf{G}_i)/J \\ \hat{\mathbf{F}}_i &= (\eta_x \mathbf{E}_i + \eta_y \mathbf{F}_i + \eta_z \mathbf{G}_i)/J \\ \hat{\mathbf{G}}_i &= (\zeta_x \mathbf{E}_i + \zeta_y \mathbf{F}_i + \zeta_z \mathbf{G}_i)/J, \end{aligned}$$

where

$$J = 1/(x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi)). \quad (3)$$

If we let ξ be the direction normal to an interface, we can transform equation (2) to characteristic form, as follows:

$$\frac{\partial \mathbf{R}}{\partial t} + \lambda \frac{\partial \mathbf{R}}{\partial \xi} = \mathbf{S}_C, \quad (4)$$

where

$$\begin{aligned}\delta\mathbf{R} &= \left(\delta\rho - \frac{\delta p}{a^2}, \delta\tilde{W}, \delta\tilde{V}, \frac{\delta p}{\rho a} + \delta\tilde{U}, \frac{\delta p}{\rho a} - \delta\tilde{U} \right)^T, \\ \lambda &= \left(U, U, U, U + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}, U - a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \right)^T.\end{aligned}$$

U is the contravariant velocity,

$$U = \xi_x u + \eta_y v + \zeta_z w, \quad (5)$$

a the speed of sound, the tilde denotes normalisation by $|\nabla\xi|$, and \mathbf{S}_C is the source term of the governing equations in characteristic form:

$$\begin{aligned}\mathbf{S}_C &= J\mathbf{P}^{-1} \left\{ \frac{\partial\hat{\mathbf{E}}_v}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_v}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_v}{\partial\zeta} \right. \\ &\quad \left. - \left[\mathbf{E}_i \frac{\partial}{\partial\xi} \left(\frac{\xi_x}{J} \right) + \mathbf{F}_i \frac{\partial}{\partial\xi} \left(\frac{\xi_y}{J} \right) + \mathbf{G}_i \frac{\partial}{\partial\xi} \left(\frac{\xi_z}{J} \right) + \frac{\partial\hat{\mathbf{F}}_i}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_i}{\partial\zeta} \right] \right\}. \quad (6)\end{aligned}$$

The matrix \mathbf{P}^{-1} (see Appendix A) transforms the conservative variables to characteristic form. The characteristic interface condition sets $\frac{\partial\mathbf{R}}{\partial t}$ equal on both sides of the interface by modifying the convective term

$$\mathbf{L} = \lambda \frac{\partial\mathbf{R}}{\partial\xi} \quad (7)$$

on one side of the interface. Which side (denoted below by L and R), is determined by the sign of the relevant component of the vector of convection speeds λ , as follows:

$$\begin{aligned}L_m^L &= L_m^R - S_{Cm}^R + S_{Cm}^L & \text{if } \lambda_m^L/|\lambda_m^L| = \lambda_m^R/|\lambda_m^R| \leq 0 \\ L_m^R &= L_m^L - S_{Cm}^L + S_{Cm}^R & \text{if } \lambda_m^L/|\lambda_m^L| = \lambda_m^R/|\lambda_m^R| \geq 0\end{aligned} \quad (8)$$

so that for each component m of \mathbf{L} , the value to be used is taken from that block with respect to which the corresponding component of λ is outgoing, and so the finite difference approximation to \mathbf{L} is calculated using one-sided differences making use only of values on that side of the interface. The

updated characteristic convection term, denoted \mathbf{L}^* , is used to update the normal convective term in equation (2) as follows:

$$\left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^* = \frac{1}{J}\mathbf{P}\mathbf{L}^* + \left[\mathbf{E}_i\frac{\partial}{\partial\xi}\left(\frac{\xi_x}{J}\right) + \mathbf{F}_i\frac{\partial}{\partial\xi}\left(\frac{\xi_y}{J}\right) + \mathbf{G}_i\frac{\partial}{\partial\xi}\left(\frac{\xi_z}{J}\right)\right], \quad (9)$$

This is the fixed-grid characteristic interface condition due to Kim and Lee [12]. To apply an interface condition of this type on a sliding plane, two modifications are necessary; firstly, the procedure requires that the source \mathbf{S}_C and convective \mathbf{L} terms be known on both sides of the interface, and as node locations in general do not conform, interpolation is required (see section 3 below). In this respect there is no difference between a sliding grid and a patched grid (in which grids on either side of interface are nonconforming, but not in motion relative to one another). The implementation of patched grids using characteristic interface conditions has already been investigated by Peers et al. [17], and Sumi et al. [26].

In addition, however, the source and convective terms need to be evaluated in the reference frame appropriate to the block to which the correction is to be applied; This reference frame is, in each case, chosen to be that in which the grid for that block is at rest, so that time derivatives of the transformation Jacobian and its terms are zero within each block. At the interface however, \mathbf{S}_C and \mathbf{L} must be calculated in both blocks and both reference frames - those values intended for use in the neighbouring block, and transformed to its reference frame, will be denoted by a leading subscript n , and we assume that we can obtain ${}_n\mathbf{Q}$ from \mathbf{Q} by Galilean transformation in physical space. Given this, it will be possible to evaluate ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$, for which the assumption of Galilean invariance is not made. These are therefore both evaluated on grids in motion with respect to their reference frames.

When evaluated on a moving grid, equation (2), and the inviscid fluxes, are modified as follows:

$$\frac{1}{J}\frac{\partial\mathbf{Q}}{\partial t} + \frac{\partial(\hat{\mathbf{E}}_i - \hat{\mathbf{E}}_v)}{\partial\xi} + \frac{\partial(\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)}{\partial\eta} + \frac{\partial(\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)}{\partial\zeta} + \mathbf{Q}\frac{\partial(J^{-1})}{\partial t} = 0. \quad (10)$$

$$\begin{aligned} \hat{\mathbf{E}}_i &= (\xi_x\mathbf{E}_i + \xi_y\mathbf{F}_i + \xi_z\mathbf{G}_i + \xi_t)/J \\ \hat{\mathbf{F}}_i &= (\eta_x\mathbf{E}_i + \eta_y\mathbf{F}_i + \eta_z\mathbf{G}_i + \eta_t)/J \\ \hat{\mathbf{G}}_i &= (\zeta_x\mathbf{E}_i + \zeta_y\mathbf{F}_i + \zeta_z\mathbf{G}_i + \zeta_t)/J, \end{aligned}$$

It is possible to demonstrate [30],[32] that numerical satisfaction of equation (10) is improved by computing J such that

$$\frac{\partial(J^{-1})}{\partial t} = - \left(\frac{\partial \hat{\xi}_t}{\partial \xi} + \frac{\partial \hat{\eta}_t}{\partial \eta} + \frac{\partial \hat{\zeta}_t}{\partial \zeta} \right). \quad (11)$$

An alternative, adopted here, is to treat the grid as fixed and set every term in (11) to zero (which certainly satisfies this identity). This is free-stream preserving (provided the numerical methods used also possess this property in the absence of grid motion), and, in an instantaneous sense, legitimate, as $\partial \mathbf{Q}/\partial t$ is Galilean invariant.

The nodal locations at which the results are obtained and later interpolated are understood to be those corresponding to the present time level t^n and not t^{n+1} , which is in any case required for explicit temporal schemes, to which we confine our discussion below. ${}_n\mathbf{Q}$, ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$ are not integrated in time in a moving reference frame; \mathbf{Q} and ${}_n\mathbf{Q}$ are calculated separately in the source block and there the latter is discarded, while the recipient's grid is stationary relative to the reference frame corresponding to ${}_n\mathbf{Q}$. $\partial(J^{-1})/\partial t$ is, therefore, never required.

${}_n\mathbf{L}$, in physical space and in the reference frame of the neighbouring block, can be obtained from the inviscid fluxes contributing to $\frac{\partial {}_n\mathbf{Q}}{\partial t}$:

$${}_n\mathbf{L} = J {}_n\mathbf{P}^{-1} \left(\xi_x \frac{\partial {}_n\mathbf{E}_i}{\partial \xi} + \xi_y \frac{\partial {}_n\mathbf{F}_i}{\partial \xi} + \xi_z \frac{\partial {}_n\mathbf{G}_i}{\partial \xi} \right). \quad (12)$$

The source term ${}_n\mathbf{S}_C$ is:

$${}_n\mathbf{S}_C = J {}_n\mathbf{P}^{-1} \left\{ \frac{\partial \hat{\mathbf{E}}_v}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}_v}{\partial \eta} + \frac{\partial \hat{\mathbf{G}}_v}{\partial \zeta} \right. \quad (13)$$

$$\left. - \left[{}_n\mathbf{E}_i \frac{\partial}{\partial \xi} \left(\frac{\xi_x}{J} \right) + {}_n\mathbf{F}_i \frac{\partial}{\partial \xi} \left(\frac{\xi_y}{J} \right) + {}_n\mathbf{G}_i \frac{\partial}{\partial \xi} \left(\frac{\xi_z}{J} \right) + \frac{\partial {}_n\hat{\mathbf{F}}_i}{\partial \eta} + \frac{\partial {}_n\hat{\mathbf{G}}_i}{\partial \zeta} \right] \right\}.$$

${}_n\mathbf{P}^{-1}$ is given in the appendix Appendix A.

Once ${}_n\mathbf{L}$ and ${}_n\mathbf{S}_C$ are obtained, and interpolated to the target grid, they are used to update \mathbf{L} as in equation (8) above, conditional on the sign of

the relevant component of λ , so as to equalise $\partial\mathbf{R}/\partial t$ on either side of the interface. The result, \mathbf{L}^* , is then used to update $\partial\hat{\mathbf{E}}_i/\partial\xi$ (equation 9), and it is only through this updated term that the sliding interface modifies equation (2); in particular, the Jacobian used to transform the LHS ($\partial\hat{\mathbf{Q}}/\partial t$) is always constant.

The correction to the original term (equation 9) is

$$\begin{aligned} \left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^* - \left(\partial\hat{\mathbf{E}}_i/\partial\xi\right) &= \frac{1}{J}\mathbf{P}({}_n\mathbf{L} - {}_n\mathbf{S}_C + \mathbf{S}_C - \mathbf{L}) \\ &= \frac{1}{J}\mathbf{P}({}_n\partial\mathbf{R}/\partial t - \partial\mathbf{R}/\partial t), \end{aligned} \quad (14)$$

That is, the correction is zero if $\partial\mathbf{R}/\partial t$ is equal on either side of the interface, as we require. It should also be zero in the case of a uniform free stream.

To verify this, we return to equation (9), which should be zero in such a case:

$$\begin{aligned} \left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^* &= \frac{1}{J}\mathbf{P}({}_n\mathbf{L} - {}_n\mathbf{S}_C + \mathbf{S}_C) \\ &+ \left[\mathbf{E}_i \left(\frac{\xi_x}{J}\right)_\xi + \mathbf{F}_i \left(\frac{\xi_y}{J}\right)_\xi + \mathbf{G}_i \left(\frac{\xi_z}{J}\right)_\xi \right], \end{aligned} \quad (15)$$

where variables with the leading prefix n are evaluated on the other side of the interface, in the same reference frame as those not so subscripted. Further, the three terms in the square brackets of equation (15) cancel their identical counterparts in equation (6), and the same is true for the equivalent terms in equation (12) for ${}_n\mathbf{L}$ and equation (13) for ${}_n\mathbf{S}_C$, in the moving reference frame, so

$$\begin{aligned} \left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^* &= \frac{1}{J}\mathbf{P} \left(J {}_n\mathbf{P}^{-1} \left[\left(\partial {}_n\hat{\mathbf{E}}_i/\partial\xi + \partial {}_n\hat{\mathbf{F}}_i/\partial\eta + \partial {}_n\hat{\mathbf{G}}_i/\partial\zeta \right) - {}_n\hat{\mathbf{S}}_v \right] \right) \\ &- \partial\hat{\mathbf{F}}_i/\partial\eta - \partial\hat{\mathbf{G}}_i/\partial\zeta + \hat{\mathbf{S}}_v \end{aligned} \quad (16)$$

Here, we have moved the viscous fluxes into the source term $\hat{\mathbf{S}}_v$ for convenience - note that they are not affected by Galilean transformations, and

${}_n\hat{\mathbf{S}}_v$ does not need to be calculated in the reference frame of the neighbouring block.

$${}_n\hat{\mathbf{E}}_i = \hat{\xi}_x {}_n\mathbf{E}_i + \hat{\xi}_y {}_n\mathbf{F}_i + \hat{\xi}_z {}_n\mathbf{G}_i, \quad (17)$$

and analogously for ${}_n\hat{\mathbf{F}}_i$, ${}_n\hat{\mathbf{G}}_i$; substituting into (16) and noting the source terms \mathbf{S}_v would be zero in a uniform flow:

$$\begin{aligned} (\partial\hat{\mathbf{E}}_i/\partial\xi)^* &= \frac{1}{J}\mathbf{P}\left(J_n\mathbf{P}^{-1}\left(\frac{\partial}{\partial\xi}\left(\hat{\xi}_x {}_n\mathbf{E}_i\right) + \frac{\partial}{\partial\xi}\left(\hat{\xi}_y {}_n\mathbf{F}_i\right) + \frac{\partial}{\partial\xi}\left(\hat{\xi}_z {}_n\mathbf{G}_i\right)\right.\right. \\ &\quad + \frac{\partial}{\partial\eta}\left(\hat{\eta}_x {}_n\mathbf{E}_i\right) + \frac{\partial}{\partial\eta}\left(\hat{\eta}_y {}_n\mathbf{F}_i\right) + \frac{\partial}{\partial\eta}\left(\hat{\eta}_z {}_n\mathbf{G}_i\right) \\ &\quad \left.\left.+ \frac{\partial}{\partial\zeta}\left(\hat{\zeta}_x {}_n\mathbf{E}_i\right) + \frac{\partial}{\partial\zeta}\left(\hat{\zeta}_y {}_n\mathbf{F}_i\right) + \frac{\partial}{\partial\zeta}\left(\hat{\zeta}_z {}_n\mathbf{G}_i\right)\right)\right) \\ &\quad - \frac{\partial\hat{\mathbf{F}}_i}{\partial\eta} - \frac{\partial\hat{\mathbf{G}}_i}{\partial\zeta} \end{aligned} \quad (18)$$

from which we obtain:

$$\begin{aligned} (\partial\hat{\mathbf{E}}_i/\partial\xi)^* &= \frac{1}{J}\mathbf{P}\left(J_n\mathbf{P}^{-1}\left(\right.\right. \\ &\quad {}_n\mathbf{E}_i\frac{\partial\hat{\xi}_x}{\partial\xi} + \hat{\xi}_x\frac{\partial{}_n\mathbf{E}_i}{\partial\xi} + {}_n\mathbf{F}_i\frac{\partial\hat{\xi}_y}{\partial\xi} + \hat{\xi}_y\frac{\partial{}_n\mathbf{F}_i}{\partial\xi} + {}_n\mathbf{G}_i\frac{\partial\hat{\xi}_z}{\partial\xi} + \hat{\xi}_z\frac{\partial{}_n\mathbf{G}_i}{\partial\xi} + \\ &\quad {}_n\mathbf{E}_i\frac{\partial\hat{\eta}_x}{\partial\eta} + \hat{\eta}_x\frac{\partial{}_n\mathbf{E}_i}{\partial\eta} + {}_n\mathbf{F}_i\frac{\partial\hat{\eta}_y}{\partial\eta} + \hat{\eta}_y\frac{\partial{}_n\mathbf{F}_i}{\partial\eta} + {}_n\mathbf{G}_i\frac{\partial\hat{\eta}_z}{\partial\eta} + \hat{\eta}_z\frac{\partial{}_n\mathbf{G}_i}{\partial\eta} + \\ &\quad \left.\left.{}_n\mathbf{E}_i\frac{\partial\hat{\zeta}_x}{\partial\zeta} + \hat{\zeta}_x\frac{\partial{}_n\mathbf{E}_i}{\partial\zeta} + {}_n\mathbf{F}_i\frac{\partial\hat{\zeta}_y}{\partial\zeta} + \hat{\zeta}_y\frac{\partial{}_n\mathbf{F}_i}{\partial\zeta} + {}_n\mathbf{G}_i\frac{\partial\hat{\zeta}_z}{\partial\zeta} + \hat{\zeta}_z\frac{\partial{}_n\mathbf{G}_i}{\partial\zeta}\right)\right) \\ &\quad - \frac{\partial\hat{\mathbf{F}}_i}{\partial\eta} - \frac{\partial\hat{\mathbf{G}}_i}{\partial\zeta} \end{aligned} \quad (19)$$

In a uniform flow, derivatives of \mathbf{Q} and the fluxes vanish, and the metric derivatives also vanish, provided ξ_x etc. vary smoothly [30]. We are left with

$$\left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^* = 0 \quad (20)$$

Recall that we use $\left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)^*$ to replace $\left(\partial\hat{\mathbf{E}}_i/\partial\xi\right)$ in equation (2), and further, that all viscous fluxes, and all other gradients of advective fluxes,

will be zero. It follows that the same will be true of $\frac{\partial \hat{\mathbf{Q}}}{\partial t}$, and therefore that a constant \mathbf{Q} will remain so.

We conclude that it is not necessary to apply a geometric conservation law for the purpose of obtaining ${}_n\mathbf{L}$ and ${}_n\mathbf{S}_C$ for subsequent interpolation. However, while the free-stream preserving property is maintained, it is possible that the method will not perform well for other flows; we show below that this is not the case.

Note that the sliding interface condition described above has been developed and tested only for inertial reference frames. In addition, as it applies only to motion tangential to the interface boundary, a more general treatment would be required for discretisations including non-periodic boundary conditions that intersect the sliding plane, and we therefore do not consider such problems below.

For a sliding or patched mesh, ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$ must be interpolated. Two options are currently implemented for interpolation - linear interpolation, and the wavenumber-optimised method of Tam and Kurbatskii [28], following the approach used for patched characteristic interfaces by Peers et al.[17].

The optimised method solves

$$\begin{aligned} \sum_{j=0}^{N-1} S_j \frac{\sin(l-j)\kappa}{(l-j)} + \frac{\lambda}{2} &= \frac{\sin(l-K+\eta)\kappa}{l-K+\eta}, l = 0, 1, 2, N-1 \\ \sum_{j=0}^{N-1} S_j &= 1, \end{aligned} \tag{21}$$

where S_j is the interpolation coefficient for the j th point of N , the size of the interpolation stencil; K is the interval number in which the interpolant is to be located (the grid points at K and $K-1$ bracket the interpolant - a centered interpolation scheme is chosen). κ is a free parameter, set, following the recommendation of the authors, to 1. This value corresponds to optimisation for accuracy at low wavenumbers; a value of 0.85, expected to perform better at higher wavenumbers, is, however, tested (see section 4.1 below). The $N+1$ equations determine the coefficients S_j and λ (not used here).

3. Implementation

For all but one of the results reported here, the interpolation described in [28] has been used, in all of those cases, with $N = 4$. Because of the

$N + 1$ equations that must be solved for every interpolation point at every Runge-Kutta substep, this has the potential to be a very costly method for a sliding grid. In this case, it is affordable, as only a single 1-D interpolation is needed per boundary node; an overset method would instead require 2-D interpolation of 2 or more halo nodes - although only of half the number of variables (${}_n\mathbf{Q}$ instead of ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$ - but equation (21) need not be solved again for those). Conversely, the overset approach becomes more attractive for inexpensive interpolation methods. Here, the following procedure is carried out at each Runge-Kutta substep:

- 1) \mathbf{S}_C and \mathbf{L} are evaluated.
- 2) ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$, their equivalents in the reference frame of the neighbouring block, are evaluated on the sliding interface (equations 13 and 22).
- 3) λ is calculated (this is independent of v_s).
- 4) Locations on the sliding plane are updated. All falling outside the original computational domain are wrapped around, and two sets of bounds calculated for the processor domain in question (wrapped and unwrapped).
- 5) Each processor searches (using a binary range tree) for neighbouring processors with which its computational domain(s) overlap.
- 6) The updated locations of grid nodes are sent to (and received from) the neighbouring process.
- 7) The nearest neighbour nodes required for interpolation are obtained (using another binary tree).
- 8) ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$ are interpolated and exchanged on sliding block interfaces.
- 9) Equation (8) is applied to correct \mathbf{L} .
- 10) The corrected \mathbf{L} is used to recalculate $\frac{\partial \mathbf{E}_i}{\partial \xi}$ (see [12]).
- 11) Less frequently (eg. at the end of a time step) the conserved variables ${}_n\mathbf{Q}$ are also interpolated and averaged. Transformation to the reference frame of the neighbour is trivial except for the total energy (pressure is interpolated instead and e_t redefined accordingly).

Note that we choose not to constrain the magnitude of the time step so that the length of the sliding interface is an integer multiple of $v_s \Delta t$, and step 4 is therefore carried out using floating point arithmetic. Because floating point arithmetic is not of unlimited precision, it is usually the case that the processor on the other side of the interface will post matching sends and receives, but it is found in practice that this cannot be relied upon (consider processor boundaries that are nearly in alignment). If it is simply assumed that neighbouring processors will always post matching receives and sends, the result is sometimes deadlock (which may happen after many tens or hundreds of thousands of time steps).

A solution to this problem is to perform step 6 using a variant of the nonblocking consensus algorithm described by Höfler et al. [9]. All sends and receives are tested for completion, a nonblocking barrier is posted once the tests are passed, and incoming messages are probed for with `MPI_Iprobe`, all in a loop which only exits if the nonblocking barrier is reached by all participating processes. This variant differs slightly from Höfler's - tests for completion take place before probing. This is because the probe (and subsequent communication) is usually unnecessary, and both `MPI_Iprobe` and `MPI_Ssend` do have overhead (see [9] for some measurements pertaining to `MPI_Ssend`).

```

done ← complete_local ← barrier_on ← false;
post sends and receives;
while not done do
  complete_local ← test whether all sends and receives complete;
  if complete_local then
    if not barrier_on then
      start nonblocking barrier;
      barrier_on ← true;
    end
    done ← test whether barrier reached by all processes;
  end
  if not done then
    msg_found ← probe all possible neighbours for incoming
    messages;
    if msg_found then
      add to list, allocate buffers, post matching receive;
    end
  end
end

```

Some optimisations are possible; for instance, the term

$$\frac{\partial_n \mathbf{E}_i}{\partial \xi} = \left(\frac{\partial \rho_n u}{\partial \xi}, \frac{\partial \rho_n u^2 + p}{\partial \xi}, \frac{\partial \rho_n u_n v}{\partial \xi}, \frac{\partial \rho_n w_n u}{\partial \xi}, \frac{\partial (\rho_n e_t + p)_n u}{\partial \xi} \right). \quad (22)$$

The normal gradient of the sliding velocity v_s , $\frac{\partial v_s}{\partial \xi}$, is zero in either reference frame (this would not be true if differentiating across the interface, but it is so for a change of (inertial) reference frame), and $\frac{\partial_n u}{\partial \xi} = \frac{\partial u}{\partial \xi}$. Derivatives of v and w are also unchanged. It is important to minimise derivative evaluations, particularly when compact differences are used, as the expense of calculating these then cannot be restricted to boundary points.

If v_s is parallel to, say, v , and $_n v = v + v_s$, then some simplifications are possible:

$$\frac{\partial_n \mathbf{E}_i}{\partial \xi} = \left(\frac{\partial \rho u}{\partial \xi}, \frac{\partial \rho u^2 + p}{\partial \xi}, \frac{\partial \rho u_n v}{\partial \xi}, \frac{\partial \rho w u}{\partial \xi}, \frac{\partial (\rho_n e_t + p) u}{\partial \xi} \right) \quad (23)$$

and, because $\frac{\partial v_s}{\partial \xi} = 0$,

$$\frac{\partial \rho u_n v}{\partial \xi} = \frac{\partial \rho u v}{\partial \xi} + v_s \frac{\partial \rho u}{\partial \xi} \quad (24)$$

This approach has been implemented in a multi-block structured curvilinear compressible Navier-Stokes solver, designed for scalability on high-performance computing systems [5]. Spatial discretisation is performed using a fourth-order central wavenumber-optimised compact finite difference scheme [13] in two spatial directions, while the third is treated spectrally. Skew-symmetric splitting is used for the convective terms [11], and a low-storage 4th-order Runge-Kutta scheme for time integration [10].

4. Verification

4.1. Vortex convection

Two 2-D inviscid vortex convection test cases were used in the first instance to verify the implementation of the sliding interface condition. The following perturbation was applied:

$$(\delta u, \delta v) = \beta e^{\ln 2(x^2 - y^2)/r^2}(\mathbf{y}, x) \quad (25)$$

$$\delta e_t = \frac{\rho_\infty}{2}(\delta u(u_\infty + \delta u) + \delta v(v_\infty + \delta v)), \quad (26)$$

with strength $\beta = 0.075$ and scaling factor $r = 5$, to free-stream values of $(\rho_\infty, u_\infty, v_\infty) = (1, 1, 0)$, and

$$e_{t\infty} = \frac{\rho_\infty}{\gamma(\gamma - 1)M^2} + \frac{\rho_\infty}{2}(u_\infty^2 + v_\infty^2), \quad (27)$$

where $M = 0.4$, $\gamma = 1.4$. The instantaneous application of the above perturbation generates an acoustic pulse in addition to the vortex, and the propagation of both the acoustic and the hydrodynamic perturbation can then be used to verify the method.

Characteristic boundary conditions [19] are applied on x -boundaries, and periodicity is assumed in y ; the domain is partitioned into four blocks, as shown in figure 2. Each block is resolved by a 128 x 128 grid and is $16r$ wide (x dimension), but trapezoidal, in order to also test the ability of the interface

condition to cope with metric discontinuities. Each y -grid line within each block is also $16r$ long, but at the sliding interface (at $x/r = 1.6$), the ξ grid lines meet at an angle of 45 degrees. In order to verify that the sliding characteristic interface performs as expected, a sliding velocity $v_s = |u_\infty| \mathbf{1}_y$ is imposed on the interface, so that the two downstream blocks move at $-v_s$ with respect to those upstream. Calculations in each block take place in the reference frame of that block, so that this motion is only expressed through the interface condition. For the downstream blocks, therefore, the initial conditions are all modified by the addition of v_s to v_∞ . Compact differencing was used as described by Kim and Sandberg [13], but no filtering was performed in this case.

Results are shown in figures 3 to 6. Figure 4 illustrates convection of vorticity, which as figure 3 shows remains almost constant; $\omega_z r / u_\infty$ is plotted against $(x - tu_\infty)/r$, at $(tu_\infty)/r = 0$ and 5, at which time the vortex has cleared the sliding interface (see figure 4 c).

Figure 6 shows p/p_∞ on $y = 0$ at $t = 25$, compared to its mirror image on the x -axis about the vortex centre at $u_\infty t$. This shows that the propagation of the pressure wave is symmetric and that it propagates at the sonic velocity a on either side of the interface. 2-D symmetry is illustrated in figure 5.

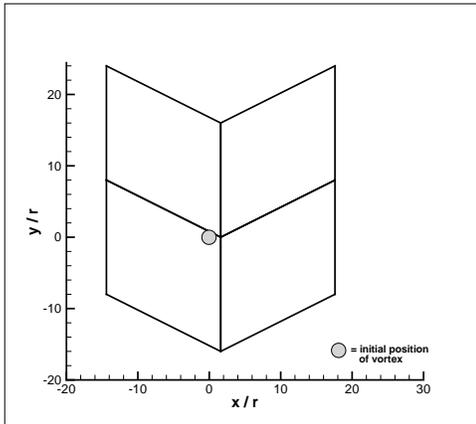


Figure 2: Vortex convection test case.

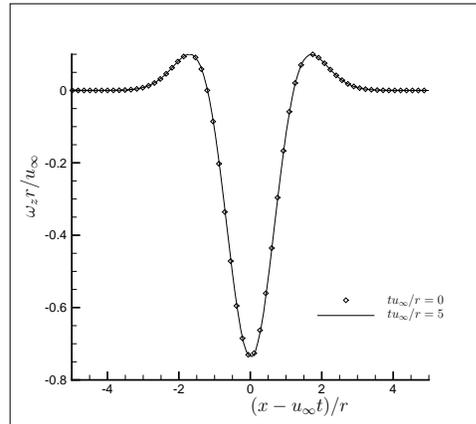


Figure 3: Vorticity at $t = 0$ (black line), and at $tu_\infty/r = 5$ (diamonds), plotted against $x - u_\infty t$.

A more quantitative analysis can be carried out by simulating an isentropic vortex. In this case, no acoustic pulse is generated, but the vortex should translate downstream unchanged. We follow Yee et al. [37] in setting

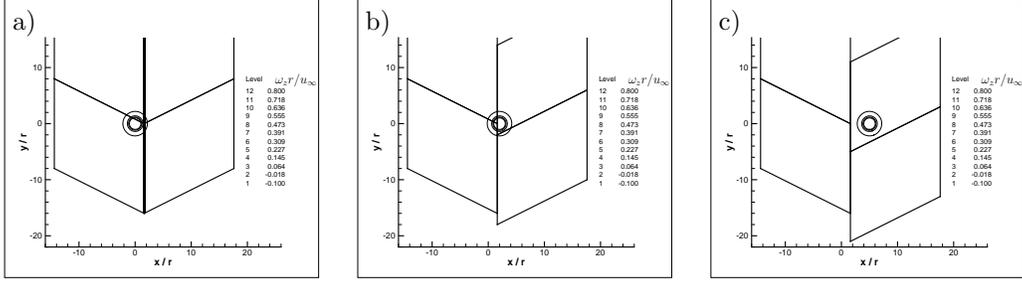


Figure 4: Vorticity contours. a) $tu_\infty/r = 0$, b) $tu_\infty/r = 2$, c) $tu_\infty/r = 5$.

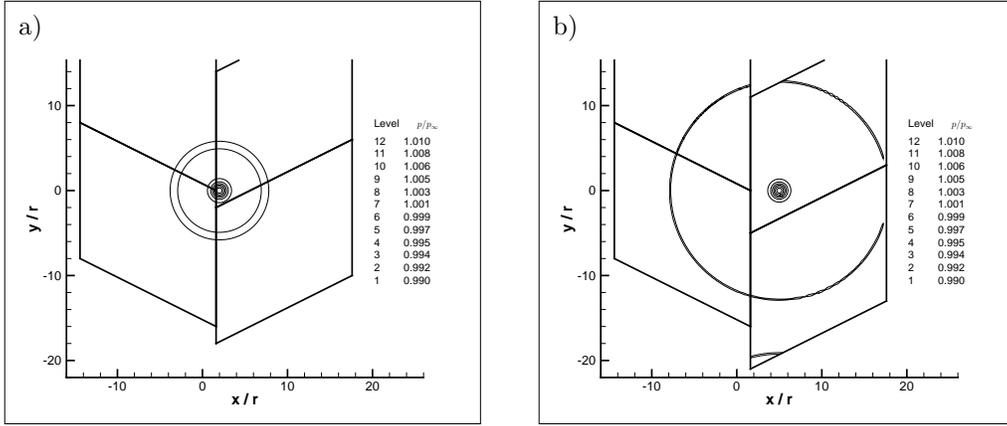


Figure 5: Pressure contours, a) $tu_\infty/r = 2$, b) $tu_\infty/r = 5$.

$$\rho = \left(1 - \frac{\beta^2(\gamma - 1)}{8\pi^2\gamma} e^{1-(x^2-y^2)/r^2} \right)^{\frac{1}{\gamma-1}}, \quad (28)$$

$$(u, v) = (u_\infty, v_\infty) + \frac{\beta}{2\pi M} e^{1/2-(x^2-y^2)/2r^2} \frac{(y, x)}{r}, \quad (29)$$

$$e_t = \frac{\rho^\gamma}{\gamma(\gamma - 1)M^2} + \frac{\rho}{2} (u^2 + v^2). \quad (30)$$

In this case, we choose $\beta = 1$, $r = 5$. Grid convergence is tested by measuring L_ρ , defined as the magnitude of the maximum error in ρ on the vortex centreline at $tu_\infty/r = 5$. This is evaluated by substituting $x - tu_\infty$ for x in equation (28). Figure 7 compares grid convergence using two interpolation methods with the fixed grid case. The grid topology illustrated in figure

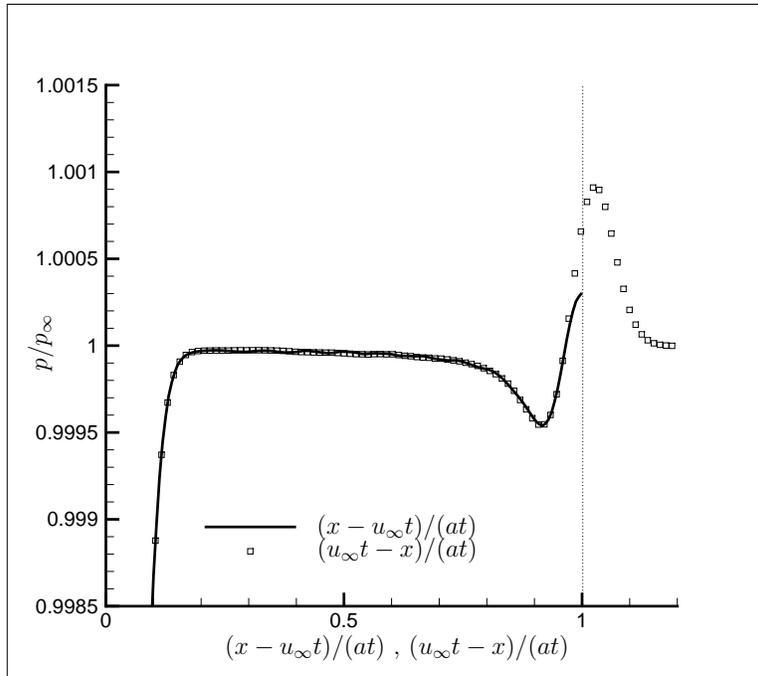


Figure 6: Pressure at $t = 25$, plotted against $(x - u_\infty t)/(at)$ (black line) and $(u_\infty t - x)/(at)$ (squares), ie. mirrored about the vortex centre (at zero on the x -axis).

2 is used in all three cases, with variable resolution. This case does not demonstrate any effect on the rate of convergence with grid refinement, either of the sliding grid, or of the two interpolation schemes, or of the parameter κ in the Tam & Kurbatskii method. This test is unfortunately, therefore, not very informative; we conclude that in this case, the sliding interface never becomes the dominant source of error.

4.2. Cylinder wake at $Re=100$

The vortex convection problem is inviscid, so, in addition, a cylinder in cross-flow at a low Reynolds number (100 based on cylinder diameter) was considered, in order to assess the new sliding-grid approach in situations where viscous effects are important. The Mach number was 0.3. Periodic boundary conditions are applied in the pitchwise direction, as in all the cases considered in this manuscript, and thus the geometry is in fact a periodic array of cylinders, separated by 20 diameters in the y direction normal to

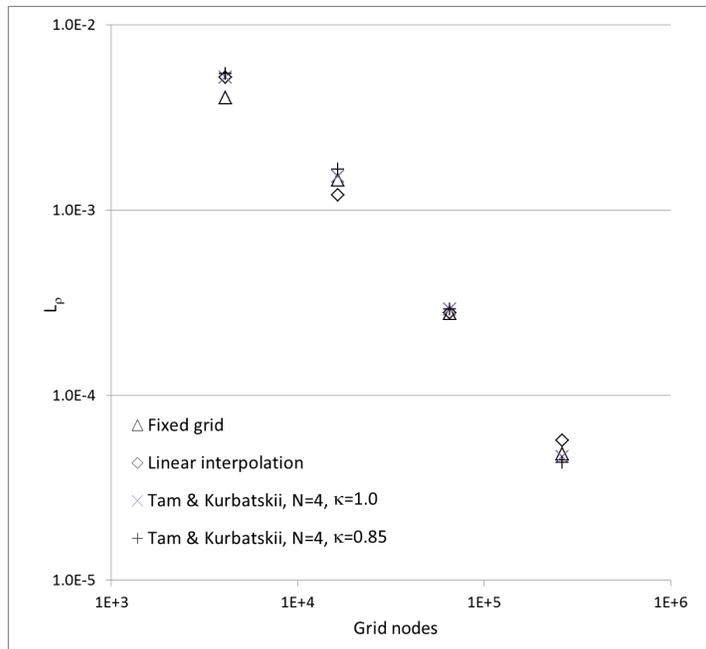


Figure 7: Grid convergence for the isentropic vortex convection case.

the free stream. Use of non-periodic boundary conditions would require *a priori* prescription of the flow at a distance from the body, and also implementation of a more general moving boundary condition than that described above, so as to treat motion normal to the boundary. Anything other than periodicity in the direction of grid motion would also require that simulations of turbulent flow use ensemble rather than time averaging to attain statistical convergence; this would increase the required computational effort substantially.

Calculations were carried out both for a fixed grid, and one including a sliding plane in the near wake of the cylinder (2.6 diameters downstream of the cylinder centre). The results presented here were obtained before the Tam & Kurbatskii scheme [28] was implemented, and thus linear interpolation was used on the sliding plane. The magnitude of the sliding velocity v_s was, in this case, $0.1U$, the free stream velocity.

The cylinder itself was simulated using an immersed boundary method, based on the continuous forcing approach of Goldstein et al. [8]; it is fully described in [34] and [22]. Characteristic boundary conditions [19] were im-

posed at the in and out flow boundaries.

The fixed grid calculation employed a grid of 420 x 200 points, stretched to maximise resolution in the vicinity of the cylinder. For the sliding calculation, the stretched grid upstream of the sliding plane was retained. However, grid stretching in y could not be used in the downstream blocks as the resolution would otherwise be inadequate much of the time. A uniform grid was necessary, and the total grid dimensions therefore increased to 300 x 200 + 120 x 800 (upstream and downstream blocks respectively). The minimum grid spacing downstream of the interface was nonetheless still coarser than upstream (see figure 8).

Vortex shedding was triggered by briefly imposing positive, then negative v at the cylinder surface (ie. transpiration). In this way, vortex shedding was initiated at precisely the same time in both calculations, so that comparison of the flow field at a given time step would be possible.

A comparison of this kind is made in figure 9, which show contours of v for both simulations, integrated to the same time level for both fixed and sliding grids. The sliding mesh simulation was repeated in 3D as an additional test; the flow remained two-dimensional, and these results were unchanged. The divergence contours shown in figure 10 shows no spurious acoustic activity at the sliding boundary (some artefacts, visible on processor boundaries shown in the figure, are considered to be the consequence of postprocessing without halo data).

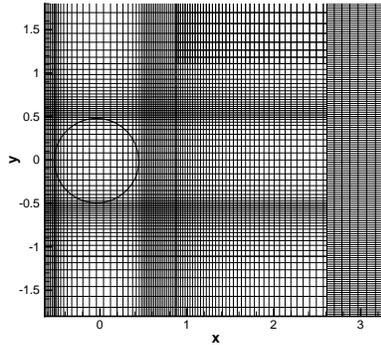


Figure 8: $Re=100$ Cylinder, mesh near the sliding plane at $x = 2.6$. Every other grid point has been omitted in both x and y .

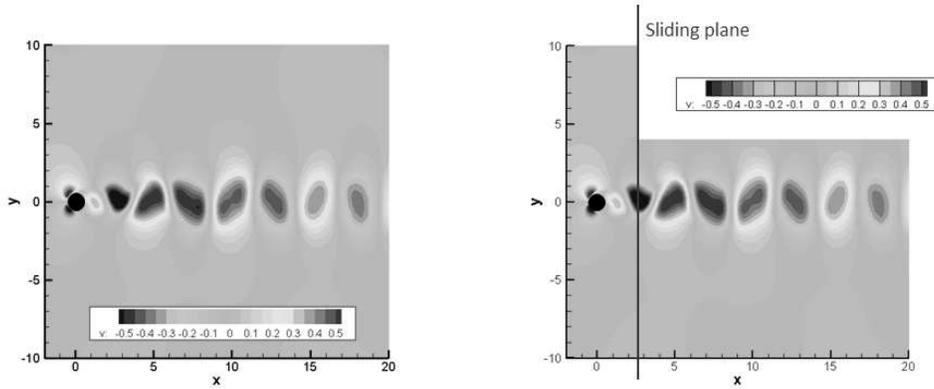


Figure 9: $Re=100$ Cylinder, v -contours on fixed and sliding meshes at the same time level.

Comparison may also be made with experimental results; the Strouhal number, based on the frequency of density fluctuations on the centreline of the cylinder wake, was found to be 0.16, in good agreement with experimental data [33] at the Reynolds number considered.

4.3. Cylinder wake at $Re=1000$

This test case was used to assess the adequacy of the method for the turbulence simulation described below in section 5. 3-D simulations were performed, of low Re ($= 1000$) and M ($= 0.3$) flow through a periodic array of cylinders separated by 20 diameters D . These parameters are similar to those of the bar wakes simulated in the application test case described in section 5 below, and the wake is expected to become turbulent, although the Reynolds number is still relatively low.

The calculations were carried out with and without a sliding interface in the near wake, on a uniform grid with a resolution of $620 \times 200 \times 66$ (in physical space) - spectral discretisation (32 dealiased Fourier modes) was used in the spanwise direction. The spanwise domain size was $10D$. A characteristic boundary condition [19] was imposed on the downstream boundary, and the cylinder once again modelled using an immersed boundary. The sliding velocity was increased to $0.2U$. The 6th-order filter of Bogey et al. [2] was used in both cases, with a filtering strength σ^{sf} of 0.2. This filter is designed to dissipate waves too short for the discretisation to represent without large

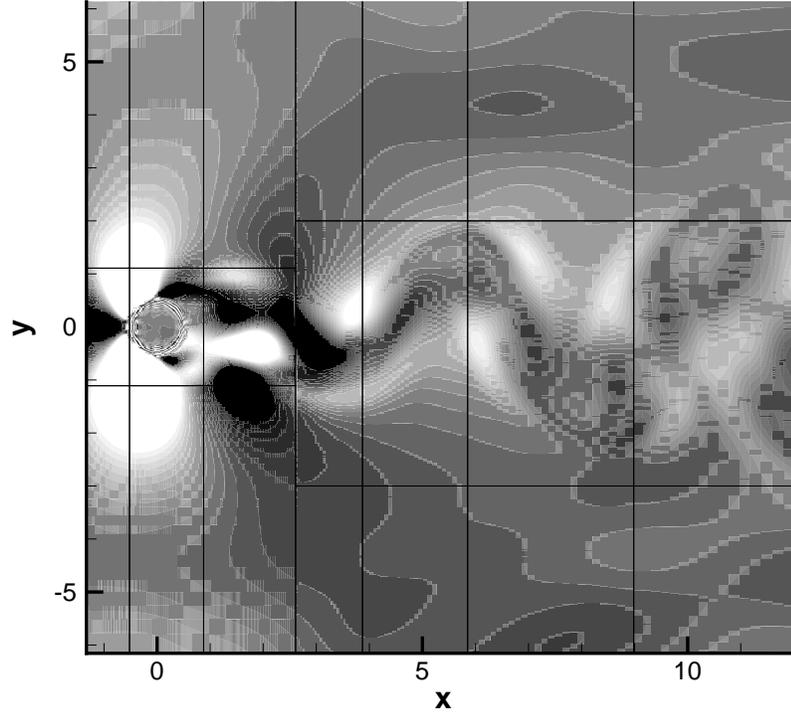


Figure 10: Re=100 Cylinder, divergence contours, sliding mesh.

dispersion errors, and has a significant effect only for $\pi/2 < k\Delta x < \pi$ (see figure 1).

Figure 11 shows the mean streamwise velocity on the wake centreline, and figures 13 and 14 the u and v components of the normal Reynolds stress (the w component was not collected). Agreement is excellent up and downstream of the interface; around the interface (at $x/D = 8$), the mean velocity is very slightly lower for the sliding mesh, and there is a small discrepancy in $\overline{u'u'}/u_\infty$. This may be due in part to the use of one-sided differencing when filtering, as it also occurs in the fixed grid results; note however that in the sliding mesh case it was necessary to collect results downstream of the interface plane by performing (2nd-order accurate) interpolation at runtime,

from the reference frame of the grid on which they were calculated to that in which the cylinder was at rest. Regardless, good agreement with fixed-grid results is seen in the wake downstream of the interface.

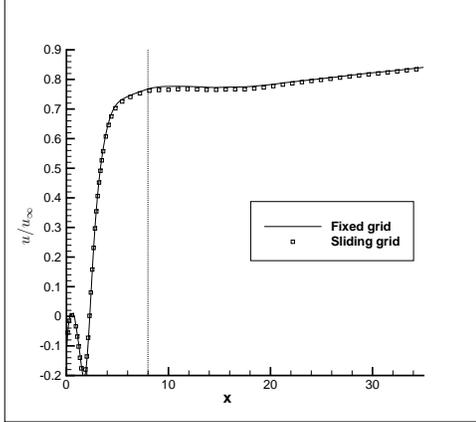


Figure 11: Re=1000 Cylinder, mean streamwise velocity on the wake centreline.

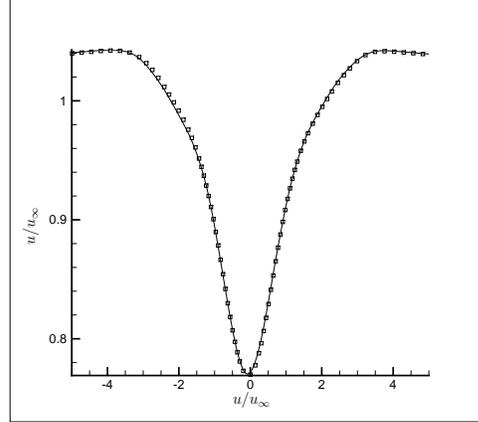


Figure 12: Re=1000 Cylinder, mean streamwise velocity profile 20 diameters downstream of the cylinder. Symbols as in figure 11.

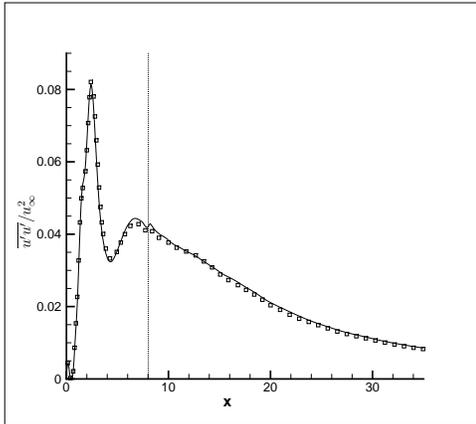


Figure 13: Re=1000 cylinder, wake centreline, $\overline{u'u'}/u_\infty^2 = (\overline{uu} - \bar{u}^2)/u_\infty^2$. Symbols as in figure 11.

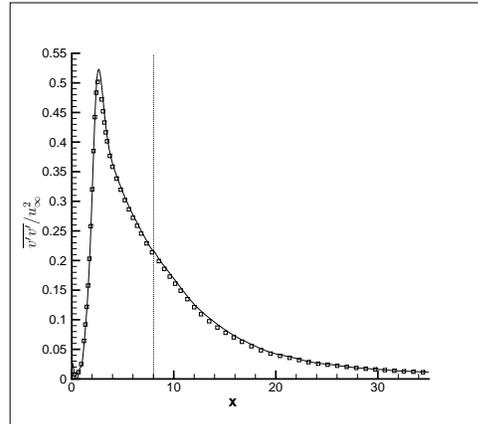


Figure 14: Re=1000 cylinder, wake centreline, $\overline{v'v'}/u_\infty^2 = (\overline{vv} - \bar{v}^2)/u_\infty^2$. Symbols as in figure 11.

Although this case was considered suitable for verifying the interface condition, the downstream boundary is too close to investigate the development of the wake; an extended grid of 63 million nodes has, however, been used to

investigate the parallel scaling of the method as part of a larger DNS calculation. Figure 15 a) shows that performance, in absolute terms (timesteps per second), is reduced slightly, in this case by between 5 and 10% at a given core count. Some of this difference can be accounted for by load imbalance; the overhead of calculating ${}_n\mathbf{S}_C$ and ${}_n\mathbf{L}$ is only borne by processes responsible for a subdomain adjacent to a sliding boundary. This has been accounted for by allocating 20% fewer grid points per core in the blocks adjoining the sliding boundary, although each of these blocks contains many processes that do not adjoin the boundary. A more sophisticated load balancing strategy should therefore improve performance at a given core count. The implied estimate of the overhead (per grid point) of the method, on adjoining processes, was arrived at by running this case with no attempt at such load balancing and comparing the measured performance with a fixed grid. Note that the optimisation represented by equation 23 was made only with respect to the spanwise direction, and that of equation 24 not at all.

Parallel scaling - as opposed to absolute performance - appears to be little affected, as shown in figure 15 b). We conclude that by assuming an increased cost of approximately 25% for processes involved in the sliding interface, and altering load balance accordingly, is sufficient to maintain performance and parallel scaling adequate for DNS of turbulent flow. This is the approach taken in section 5 below.

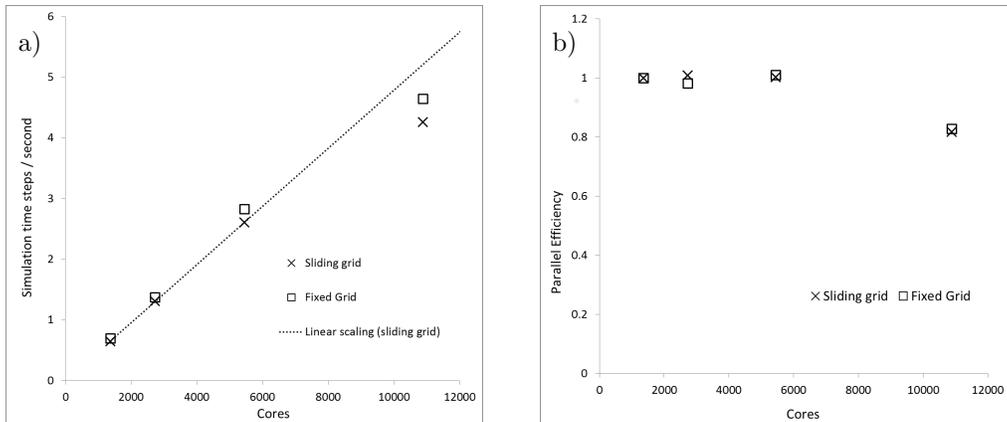


Figure 15: a) - Performance (time steps per second) scaling of sliding and fixed grids of ≈ 64 million points each. b) - Parallel efficiency, measured in each case relative to performance at 1360 cores.

5. Application

We apply the method to simulate a single-stage low pressure turbine cascade subject to inflow disturbances created by an upstream array of cylinders moving in the pitchwise direction relative to the blades. The geometry is essentially that investigated experimentally by Stadtmüller et al., [23], as modified by Wu and Durbin [35] and Michelassi et al. [16] in their numerical studies - that is, the bar to blade pitch ratio is set to an integer number to permit the use of periodic boundary conditions in the pitchwise direction, which is desirable for numerical simulation. This case, and variations of it, are also the subject of [18] and [15], where, however, a moving immersed boundary condition is used. Flow and computational parameters are listed in table 1 below.

Isentropic Reynolds number CV_{2is}/ν	60000
Isentropic Exit Mach number V_{2is}/a	0.4
Ratio of specific heats $\gamma = c_p/c_v$	1.4
Pr	0.72
Pitch/Chord ratio y_P/C	0.799
Inlet flow angle	46.05°
Stagger angle	30.70°
Axial chord length C_{ax}	0.86 C
Blade-Bar separation	0.7 C
Bar diameter	0.02 C
Bar pitch	$y_P/2$, 19.98 D
Bar velocity	0.41 U_{ref}
Reduced frequency fC/V_{2is}	0.61
Inflow turbulence intensity	0.0%
Total grid points	2.1x10 ⁷
Computational domain width	0.2 C
Time step (fixed grid)	2.537x10 ⁻⁵ C/U_{ref}
Time step (sliding grid)	5.075x10 ⁻⁵ C/U_{ref}

Table 1: Simulation parameters.

The domain is partitioned into blocks as shown in figure 16, such that the 4 blocks nearest the blade form an O-grid, and with the sliding plane located at a block interface at $x/C = -0.34$, where the leading edge of the blade is at $x/C = 0$.

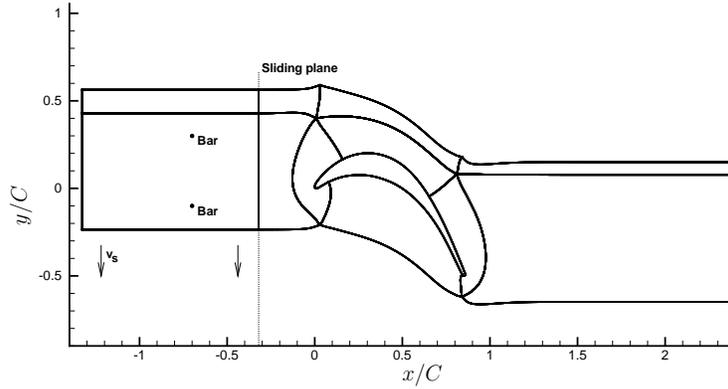


Figure 16: Block topology.

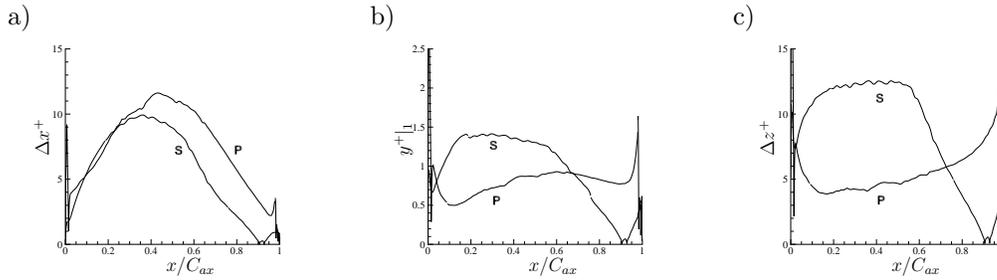


Figure 17: Grid resolution in wall units (P=pressure side, S=suction side of the blade); a) Streamwise resolution, b) height of first grid node above the blade surface, c) spanwise resolution.

Fairly adequate resolution can be maintained over both surfaces of the airfoil. Figure 17 shows the resolution in wall units (eg. $\Delta x^+ = \Delta x u_\tau / \nu$); this is of course variable over the surface; excursions from acceptable norms occur over the forward half of the suction side (where, however, the boundary layer is laminar), and near the trailing edge on the pressure side (where large streamwise-aligned vortices occur, but do not trigger transition). The cylinder is less well resolved, and should be regarded as a wake generator rather than as part of the simulation. It is again modelled using the immersed boundary method described in [34] and [22].

A simulation using a fixed grid and moving immersed boundary was also carried out. In this case the immersed boundary points were moved every

time step (see [22]). Both approaches are considered viable for this simulation, which does not require fine resolution of the moving cylinders - for a rotor-stator interaction study involving direct simulation of the upstream blade row, however, the moving immersed boundary requires that the near-wall resolution be maintained throughout the computational domain, and this may not be affordable for DNS or wall-resolved LES. Figure 18 illustrates the qualitative similarity of the two simulations - here, two instantaneous snapshots are chosen. The flow is turbulent, and thus quantitative comparisons cannot be made as in figure 9 above, but only through comparison of time averaged quantities.

Both simulations were run for 15-20 passes of bar over blade before collecting statistics (because there are two bars, this corresponds to only half that number of full domain cycles). Reported results were averaged over at least 20 bar passes. This was sufficient for statistical convergence of the quantities examined.

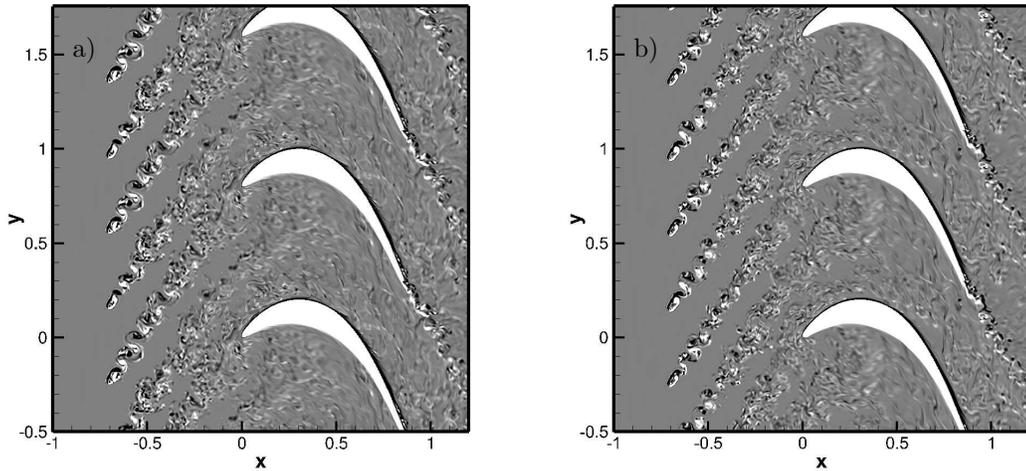


Figure 18: Instantaneous spanwise vorticity contours, a) moving immersed boundary method, b) sliding grid.

With the combination of parameters given in table 1, the boundary layer on the suction side of the blade is observed to separate in the absence of bar wakes, but to remain attached when they are present, owing to bypass transition triggered by bar wake turbulence [35], [16]. This behaviour is re-

produced in both simulations (sliding mesh and moving immersed boundary), as is illustrated by the static pressure coefficient c_p on the blade shown in figure 19, where c_p is

$$c_p = \frac{p_w - \overline{p_2}}{\overline{p_{t1}} - \overline{p_2}}. \quad (31)$$

Here p_w is the static pressure at a point on the surface of the blade, and $\overline{p_{t1}}$ and $\overline{p_2}$ are, respectively, the mass-flow weighted spatial averages of upstream total and downstream static pressure. The ‘upstream’ point 1 corresponds to $x = -0.3C$, upstream of the leading edge (but downstream of the bars), and averaging is performed over the y coordinate; the same approach is used for the ‘downstream’ point 2, which is at $0.344C$ downstream of the trailing edge.

The wake kinetic loss coefficient c_ω , which is given by

$$c_\omega = \frac{\overline{p_{t1}} - p_{t2}}{\overline{p_{t1}} - \overline{p_2}}, \quad (32)$$

(where p_{t2} is the mass-flow weighted average total pressure at on the plane on which ω is plotted) downstream of the blade is shown in figure 20, and this is also in reasonably good agreement. This coefficient is not defined consistently in the literature, and sensitive to the type of averaging used (we have used mass-flow weighted averages), so we do not present comparisons with other published results here.

Parallel scaling is fair for this case, considering the relatively small number of grid points (approx. 20 million) for a DNS. This is shown in figure 21, where an attempt has been made to compare this method with the use of a moving immersed boundary to achieve a similar result. The code implements hybrid MPI/OMP parallelism, so OMP scaling is shown for two MPI decompositions, which agree fairly well where they overlap in core count. This is also the case for the moving immersed boundary, which is therefore represented by a single line; mixed OMP and MPI scaling is used, with similar core counts as for the sliding mesh. The measurements were taken on an IBM Blue Gene/Q system.

6. Discussion

A method of implementing a sliding grid using a characteristic interface condition is developed, and verified using a set of test cases relevant to direct

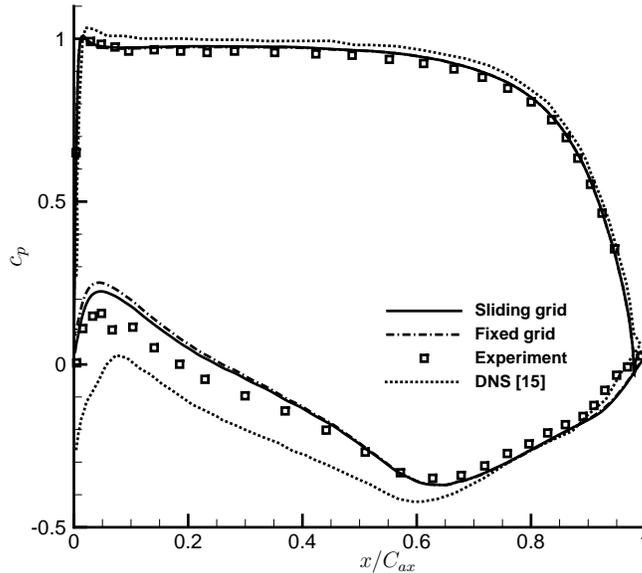


Figure 19: Static pressure coefficient c_p . Fixed grid results obtained using a moving immersed boundary. The DNS results of Michelassi are described in [16]. Experimental results are due to Stadtmüller et al., [23] (extracted from [16]).

numerical simulation of turbulent flow in moderately complex geometries. Accuracy and performance in a high performance computing context are demonstrated to be adequate. This method is applied to simulate cylinder wakes impinging on the T-106A geometry at a low Reynolds number. These wakes affect transition and separation of the suction-side boundary layer.

The sliding interface and moving immersed boundary methods described and tested in this paper agree quite closely. The blade pressure distribution (figure 19) is, in both cases, in reasonably good agreement with the experimental results of Stadtmüller et al. [23]. In particular, the suppression of the separation bubble near the trailing edge on the suction side (which occurs when the incoming wakes are removed) is captured. Perhaps surprisingly, the experimental results are, on the suction side, bracketed by the DNS results of Michelassi et al. [16] and the current results. However, the DNS described in [16] was incompressible, and significant uncertainty exists in relation to

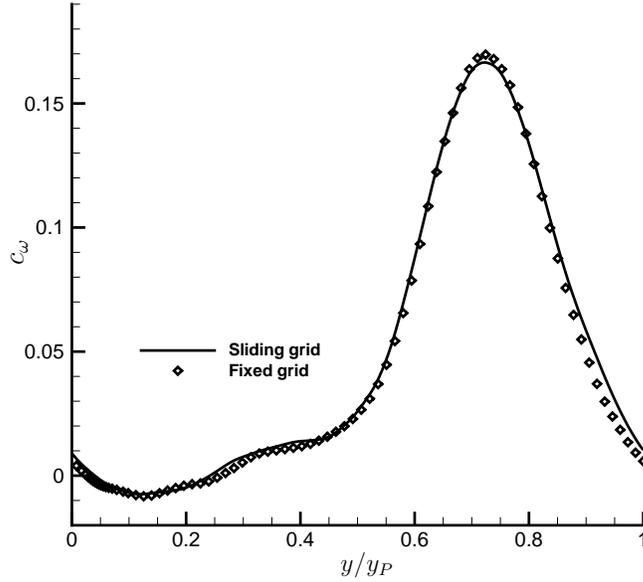


Figure 20: Total pressure or kinetic loss coefficient $c_\omega = \frac{p_{t1} - p_{t2}}{p_{t1} - p_K}$, at $x/C_{ax} = 0.4$ downstream of the trailing edge.

the inflow angle that should be used. 37.7° was intended by Stadtmüller, but the actual angle was estimated at 45.5° on the basis of RANS simulations [16]. This value was used by Michelassi et al., who also performed RANS simulations suggesting a different inflow angle would improve agreement of DNS and experiment.

Parallel scaling (shown in figure 21) is considered acceptable for the problem considered. The need to compute ${}_n\mathbf{L}$ and ${}_n\mathbf{S}_C$ introduces some overhead, particularly evident at low core counts, but this overhead is purely computational and parallelises very well, providing it is properly load balanced. Note that in that particular case it proved possible to use a larger time step for the sliding grid simulation; this has been ignored for the purpose of this comparison.

The domain decomposition for the moving immersed boundary uses subdomains of identical size; that used for the sliding grid reduces their size in blocks adjacent to the sliding interface to improve load balance. This

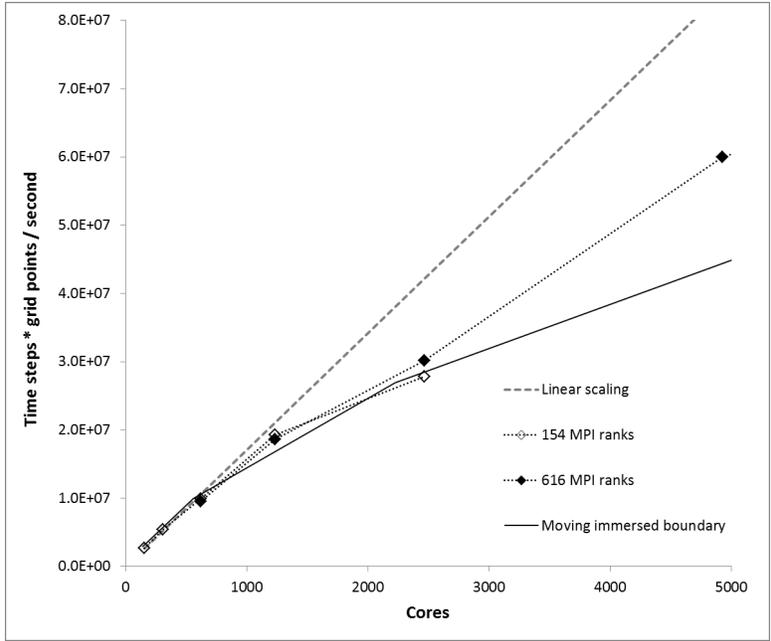


Figure 21: Scaling for bar wake / T106 blade interaction simulation, sliding grid and moving immersed boundary.

seems like a plausible explanation for the difference in scaling at higher core counts, but use of the sliding grid decomposition with the immersed boundary method did not result in an improvement. The overhead due to the immersed boundary also introduces load imbalance, but this is present in both approaches - the overhead of moving the boundary, however, is present only in the moving immersed boundary method. This is difficult to mitigate further as the overhead is not tied to particular grid nodes (unlike the overhead due to a static immersed boundary or a sliding mesh). Presumably dynamic load balancing would be required, but has not been considered here as the focus is on the sliding grid approach.

Discussion of the results in the context of rotor-stator interaction in turbomachinery flow is deferred to a future study to which, it is hoped, they will contribute. The good agreement attained between the two methods - the moving immersed boundary and the sliding grid - serves, in conjunction with the test cases of section 4, to give confidence in the validity of the method.

7. Acknowledgements

This work was partially funded by GE Global Research, and supported by the UK Turbulence Consortium, funded by the EPSRC under grant EP/G069581/1. Use was made of the facilities of HECToR, the UK's national high-performance computing service, and of the STFC Hartree Centre.

References

- [1] BOGEY, C., AND BAILLY, C. A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *Journal of Computational Physics* 194 (2004), 194–214.
- [2] BOGEY, C., DE CACQUERAY, N., AND BAILLY, C. A shock-capturing methodology based on adaptative spatial filtering for high-order non-linear computations. *Journal of Computational Physics* 228 (2009), 1447–1465.
- [3] CUI, J., RAO, V., TUCKER, P., AND JEFFERSON-LOVEDAY, R. Numerical investigation of separated boundary layer transition in high-lift low pressure turbines. *unknown AIAA* (2013).
- [4] DAVIS, R., YAO, J., CLARK, J., STETSON, G., ALONSO, J., A.JAMESON, HALDEMAN, C., AND DUNN, M. Unsteady interaction between a transonic turbine stage and downstream components. *International Journal of Rotating Machinery* 10 (2004), 495–506.
- [5] EDWARDS, T., AND SANDBERG, R. Parallelising HipSTAR using OpenMP. Tech. rep., CRAY Centre of Excellence Project report, 2011.
- [6] FENWICK, C., AND ALLEN, C. Development and validation of sliding and non-matching grid technology for control surface representation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 220 (2006), 220–299.
- [7] GAGNON, J.-M., CIOCAN, G., DESCHENES, C., AND ILIESCU, M. Numerical and experimental investigation of rotor-stator interactions in an axial turbine : numerical interface assesement. In *Proceedings of the ASME Fluids Engineering Division Summer Meeting* (Jacksonville, Florida, August 2008). FEDSM2008-55183.

- [8] GOLDSTEIN, D., HANDLER, R., AND SIROVICH, L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics* 105 (1993), 354–366.
- [9] HÖFLER, T., SIEBERT, C., AND LUMSDAINE, A. Scalable communication protocols for dynamic sparse data exchange. In *2010 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'10)* (2010), ACM, pp. 159–168.
- [10] KENNEDY, C., CARPENTER, M., AND LEWIS, R. Low-storage, explicit runge-kutta schemes for the compressible navier-stokes equations. *Applied Numerical Mathematics* 35 (2000), 177–219.
- [11] KENNEDY, C., AND GRUBER, A. Reduced aliasing formulations of the convective terms within the navier-stokes equations for a compressible fluid. *Journal of Computational Physics* 227 (2008), 1676–1700.
- [12] KIM, J., AND LEE, D. Characteristic interface conditions for multiblock high-order computation on singular structured grid. *AIAA Journal* 41 (2003), 2341–2348.
- [13] KIM, J., AND SANDBERG, R. Efficient parallel computing with a compact finite difference scheme. *Computers and Fluids* 58 (2012), 70–87.
- [14] LELE, S. Compact finite-difference schemes with spectral-like resolution. *Journal of Computational Physics* 103 (1992), 16–42.
- [15] MICHELASSI, V., CHEN, L., PICHLER, R., AND SANDBERG, R. Compressible direct numerical simulation of low-pressure turbines: Part ii - effect of inflow disturbances. In *Proceedings of the 2014 ASME IGTI Turbo Expo Conference* (Düsseldorf, Germany, June 2014). GT2014-25689.
- [16] MICHELASSI, V., WISSINK, J., AND RODI, W. Analysis of DNS and LES of flow in a low pressure turbine cascade with incoming wakes and comparison with experiments. *Flow, Turbulence, and Combustion* 69 (2002), 295–330.
- [17] PEERS, E., ZHANG, X., AND KIM, J. Patched characteristic interface condition for high-order multiblock aeroacoustic computation. *AIAA Journal* 48 (2010), 2512–2522.

- [18] PICHLER, R. S. R., CHEN, L., JOHNSTONE, R., AND MICHELASSI, V. Compressible direct numerical simulation of low-pressure turbines: Part i - methodology. In *Proceedings of the 2014 ASME IGTI Turbo Expo Conference* (Düsseldorf, Germany, June 2014). GT2014-25685.
- [19] POINSOT, T., AND LELE, S. Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics* 101 (1992), 104–129.
- [20] RAI, M. An implicit, conservative, zonal-boundary scheme for Euler equation calculations. *Computers and Fluids* 14 (1986), 295–316.
- [21] RAI, M. A direct numerical simulation of turbine rotor-stator interaction. In *39th AIAA Fluid Dynamics Conference* (San Antonio, Texas, June 2009). AIAA-2009-3685.
- [22] SCHLANDERER, S., AND SANDBERG, R. DNS of a compliant trailing-edge flow. In *19th AIAA/CEAS Aeroacoustics Conference* (Berlin, Germany, May 2013). AIAA-2013-2013.
- [23] STADTMÜLLER, P. Investigation of wake-induced transition on the lp turbine cascade T106 A-EIZ. Tech. rep., University of the Armed Forces, Munich, 2001.
- [24] STEGER, J., AND BENEK, J. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering* 64 (1987), 301–320.
- [25] STEIJL, R., AND BARAKOS, G. Sliding mesh algorithm for CFD analysis of helicopter rotor-fuselage aerodynamics. *International Journal for Numerical Methods in Fluids* 58 (2008), 527–549.
- [26] SUMI, T., KUROTAKI, T., AND HIYAMA, J. Interpolated characteristic interface conditions for zonal grid refinement of high-order multi-block computations. *International Journal of Computational Fluid Dynamics* 26 (2012), 23–43.
- [27] TAM, C. Computational aeroacoustics: Issues and methods. *AIAA Journal* 33 (1995), 1788–1796.

- [28] TAM, C., AND KURBATSKII, K. A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes. *Journal of Computational Physics* 147 (2000), 588–617.
- [29] TAM, C., AND WEBB, J. Dispersion-relation-preserving finite difference schemes for computational aeroacoustics. *Journal of Computational Physics* 107 (1993), 262–281.
- [30] THOMAS, P., AND LOMBARD, C. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* 17 (1979), 1030–1037.
- [31] TYAGI, M., AND ACHARYA, S. Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method. *International Journal for Numerical Methods in Fluids* 48 (2005), 691–722.
- [32] VISBAL, M., AND GAITONDE, D. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics* 150 (1999), 199–238.
- [33] WILLIAMSON, C. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics* 28 (1996), 477–539.
- [34] WINKLER, J., SANDBERG, R., AND MOREAU, S. Direct numerical simulation of the self-noise radiated by an airfoil in a narrow stream. In *18th AIAA/CEAS Aeroacoustics Conference (33rd AIAA Aeroacoustics Conference)* (Colorado Springs, Colorado, June 2012). AIAA-2012-2059.
- [35] WU, X., AND DURBIN, P. Evidence of longitudinal vortices evolved from distorted wakes in a turbine passage. *Journal of Fluid Mechanics* 446 (2001), 199–228.
- [36] WU, X., JACOBS, R., HUNT, J., AND DURBIN, P. Simulation of boundary layer transition induced by periodically passing wakes. *Journal of Fluid Mechanics* 398 (1999), 109–153.
- [37] YEE, H., SANDHAM, N., AND DJOMEHRI, M. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics* 150 (1999), 199–238.

- [38] YEOH, S., PAPADAKIS, H., AND YIANNESKIS, M. Determination of mixing time and degree of homogeneity in stirred vessels with large eddy simulation. *Chemical Engineering Science* 60 (2005), 2293–2302.

Appendix A. Inverse transformation matrix

The transformation matrix \mathbf{P} , multiplication with which is used to convert the corrected flux \mathbf{L} from characteristic to conserved variables (given in [12]) does not require modification when interpolation is carried out in the reference frame of the recipient block. Its inverse - which is needed to calculate ${}_n\mathbf{L}$ and ${}_n\mathbf{S}_C$, however, does. It is given by:

$${}_n\mathbf{P}^{-1} = \begin{pmatrix} {}_n\mathbf{B}_0 \cdot \mathbf{l}_x & \frac{(\gamma-1)}{a^2} {}_n u \tilde{\xi}_x & \frac{(\gamma-1)}{a^2} {}_n v \tilde{\xi}_x + \frac{\tilde{\xi}_z}{\rho} & \frac{(\gamma-1)}{a^2} {}_n w \tilde{\xi}_x + \frac{\tilde{\xi}_y}{\rho} & -\frac{\gamma-1}{a^2} \tilde{\xi}_x \\ {}_n\mathbf{B}_0 \cdot \mathbf{l}_y & \frac{(\gamma-1)}{a^2} {}_n u \tilde{\xi}_y - \frac{\tilde{\xi}_z}{\rho} & \frac{(\gamma-1)}{a^2} {}_n v \tilde{\xi}_y & \frac{(\gamma-1)}{a^2} {}_n w \tilde{\xi}_y + \frac{\tilde{\xi}_x}{\rho} & -\frac{\gamma-1}{a^2} \tilde{\xi}_y \\ {}_n\mathbf{B}_0 \cdot \mathbf{l}_z & \frac{(\gamma-1)}{a^2} {}_n u \tilde{\xi}_z - \frac{\tilde{\xi}_y}{\rho} & \frac{(\gamma-1)}{a^2} {}_n v \tilde{\xi}_z + \frac{\tilde{\xi}_x}{\rho} & \frac{(\gamma-1)}{a^2} {}_n w \tilde{\xi}_z & -\frac{\gamma-1}{a^2} \tilde{\xi}_z \\ \frac{a}{\rho} \left(\frac{\gamma-1}{2} {}_n M^2 - \frac{{}_n \mathbf{v} \cdot \mathbf{l}_\xi}{a} \right) & {}_n \mathbf{C}_+ \cdot \mathbf{l}_x & {}_n \mathbf{C}_+ \cdot \mathbf{l}_y & {}_n \mathbf{C}_+ \cdot \mathbf{l}_z & \frac{\gamma-1}{\rho a} \\ \frac{a}{\rho} \left(\frac{\gamma-1}{2} {}_n M^2 - \frac{{}_n \mathbf{v} \cdot \mathbf{l}_\xi}{a} \right) & {}_n \mathbf{C}_- \cdot \mathbf{l}_x & {}_n \mathbf{C}_- \cdot \mathbf{l}_y & {}_n \mathbf{C}_- \cdot \mathbf{l}_z & \frac{\gamma-1}{\rho a} \end{pmatrix} \quad (\text{A.1})$$

where

$$\mathbf{B}_0 = \left(1 - \frac{\gamma-1}{2} {}_n M^2 \right) \mathbf{l}_\xi - \frac{1}{\rho} ({}_n \mathbf{v} \times \mathbf{l}_\xi) \quad (\text{A.2})$$

$${}_n \mathbf{C}_\pm = \pm \frac{\mathbf{l}_\xi}{\rho} - \frac{\gamma-1}{\rho a} {}_n \mathbf{v} \quad (\text{A.3})$$