Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona, Spain, July 9-13, 2018 **ICCFD10-2018-088**

Transitional shockwave/boundary-layer interactions in the automatic source-code generation framework OpenSBLI

D.J. Lusher*, S.P. Jammy* and N.D. Sandham*
Corresponding author: d.lusher@soton.ac.uk

* Aerodynamics and Flight Mechanics group, University of Southampton
Southampton, SO17 1BJ, United Kingdom.

Abstract: Shockwave/boundary-layer interactions (SBLI) are an important design consideration for many aeronautical flows, including supersonic engine intakes and transonic wings. In many simulations of these phenomena span-periodicity is assumed, neglecting complex flow phenomena resulting from physical flow confinement. In this paper 3D SBLI with sidewall confinement effects are simulated for laminar and transitional boundary layers, in the automatic source code generation framework OpenSBLI. An incident shock of wave angle 31.6° impinges on a laminar boundary layer for a Mach 2 inlet. The incident shock interacts with both the sidewall and bottom wall boundary layers, causing large regions of flow reversal. A laminar base flow is obtained, which is then forced to transition by the introduction of upstream disturbances. A 6th order Targeted Essentially non-Oscillatory (TENO) scheme is used for shock capturing. OpenSBLI generates code in the domain specific language of the Oxford Parallel Structured software (OPS), which is then translated to a range of parallel programming paradigms. All computations were performed on multi-GPU configurations of NVIDIA P100/V100 GPUs.

Keywords: Shockwave, SBLI, WENO, TENO, GPU, Code-generation, Finite-difference

1 Introduction: shockwave/boundary-layer interactions

The study of shockwave/boundary-layer interactions (SBLI) is important in the context of aeronautical design, with common applications including but not limited to: transonic aerofoils, supersonic engine intakes, and over-expanded nozzles [1]. SBLIs have been an active area of research for the past 70 years [2]. The review by [2] highlights the need for an improved understanding of unsteady shock motion and complex three-dimensional (3D) effects. In the two decades since that review, significant improvements have been made in the available computational power, which now enables unsteady 3D SBLI effects to be investigated numerically. Shock interactions with laminar boundary-layers are of interest because they can provide information on shock-induced transitions to turbulence. A common starting point for numerical study of shock induced separation of a laminar boundary-layer is the work by [3], in which 2D simulations were performed for SBLI with Mach numbers ranging from 1.4 to 3.4. Based on the experiments of [4], the numerical results were largely in good agreement with experimental values despite an over estimation of the size of the separated region. This discrepancy may be due to 3D effects in the experiments that were not present in the 2D simulation. The parameters influencing the length of the separation bubble were investigated, with a linear scaling of the length with increasing shock strength observed. A recent review of this study by [5] examined some of the issues with the original work via 2D direct numerical simulation (DNS), notably the need for a large integration time for a fully developed separation bubble.

The review by [1] discusses the need for better understanding of three-dimensional effects for SBLIs. Central to this aim is the addition of sidewall effects that are present in most realistic geometries, and the need for more extensive parametric studies to investigate whether current findings remain valid at higher Reynolds numbers. Internal flows contain substantial complexity, as multiple shocks interact and reflect between the walls of the enclosed environment. At the corner between solid walls conical shocks are believed to form and interact with the initial incident shock [6]. The presence of a sidewall is known to cause a curving of the incident shock compared to quasi-2D periodic cases, and can strengthen the shock and corresponding central separation bubble [7].

A recent numerical study [7] performed an LES for a turbulent SBLI with sidewall effects at Mach 2.7. In addition to the central separation region on the bottom wall, two smaller corner separations were observed. Between the regions of corner separation and central separation, a region of attached flow was seen on both sides. By reducing the aspect ratio from four to one a shift upstream of the separation and reattachment points of the central separation was observed compared to the quasi-2D results. The length of the separation bubble was increased by 30% compared to the quasi-2D results, and a curving of the incident shock was seen. Interaction between the incident shock and the sidewall boundary-layer caused three-dimensional effects for the central separation that are not present in quasi-2D studies. Notably near the sidewall, the incident shock is weakened from compression waves resulting from the shock-induced thickening of the sidewall boundary-layer. Further away from the wall the main shock becomes curved and is strengthened. Due to the increased susceptibility of laminar boundary-layers to separate, 3D SBLI of laminar boundary-layers with a sidewall are expected to exhibit much larger regions of separation compared to the turbulent case.

A detached eddy simulation (DES) of a full wind tunnel configuration with turbulent boundary-layer was performed by [8]. It was observed that the presence of sidewalls reduced the effective section of the wind tunnel geometry, and led to a strengthening of the main interaction. Strong fluctuations were found in the corner regions, and a low frequency motion of the reflected shock was observed. Wall-modelled LES of oblique SBLI with turbulent boundary-layers was performed by [9], to study confinement effects present in rectangular duct SBLIs. Strong three-dimensionality was seen in the mean separation due to the confinement effects, and a larger separation bubble was observed relative to a span-periodic 3D case. A train of reflected oblique shocks was captured, causing secondary separation on the upper surface of the domain.

Several experimental studies have been performed to investigate sidewall effects on SBLIs, which often include the use of flow controls to reduce the onset and extent of the separation. The work of [10] studied corner flows in rectangular supersonic inlets for Mach numbers of 2.0 and 2.75. These exploratory experiments used wedges of 6° and 10° to generate the incident shock, and noted the difficulty there is in matching simulation and experiment in this area. Recirculation in the corner region was observed to move upstream and form secondary shocks that interacted with the main shock, and in several configurations unstart was observed. They noted that cases with more moderate separation were better suited to computational fluid dynamics (CFD), as disturbances would not propagate upstream to the leading edge; reattachment would also occur more promptly to avoid separation induced issues with outflow boundary conditions.

Experimental studies [6],[11], have investigated turbulent oblique SBLI within rectangular channels at Mach 2.5. The aim of the experiments was to observe and manage the onset and magnitude of the corner separation. It was observed that three-dimensional effects were not confined to corner regions, since oil streak patterns identified three-dimensionality throughout the channel. The shape and size of the central separation region was altered significantly from a quasi-2D configuration, a feature that was most prominent for exaggerated corner separations. Conical shocks emanating at the onset of corner separation affected the adverse pressure gradient felt in other regions of the flow, further highlighting the shortcomings of quasi-2D (span-periodic) studies of SBLI. Various control methods have been applied in the corner region in an attempt to suppress the three-dimensionality introduced to the flow by corner separation. The work of [12] used wind tunnel experiments to assess the impact of rectangular vortex generators on regions of corner separation. For a Mach 1.4 inlet, the experiments studied the interaction of a turbulent boundary-layer with a normal shock-wave. Vortex generators were shown to be effective at reducing the observed separation in the corner region, but large regions of separation remained in the centre of the duct.

1.1 Code generation and domain specific languages

Scientific codes are often developed over several decades, containing significant portions of legacy code that can be incompatible with emerging computational architectures and modern software practices. Codes written for conventional CPU nodes with MPI are often difficult to port to GPU and many-core accelerators, requiring non-trivial development and validation efforts. In addition to the insight into laminar SBLI with confinement effects, the present work demonstrates the progress made in the development of OpenSBLI [13] [14], a Python-based source code generator for scientific applications. OpenSBLI is a platform to generate code to solve systems of partial differential equations via finite-difference methods on structured meshes, starting from an easy to use high-level Python interface. Although OpenSBLI is designed to be able to handle a wide range of differential equations, its main application to date has been solving the compressible Navier-Stokes equations. OpenSBLI takes user specified equations, expands and discretizes them according to the selected numerical schemes, outputting a C code with all of the components required to solve the problem.

The C code is written in the domain specific language (DSL) of the Oxford Parallel Structured (OPS) software library [15], that takes a base sequential code and produces parallel codes for a variety of common parallel programming paradigms. The coupling of OpenSBLI with OPS is known as a 'separation of concerns': the physical problem and numerical algorithms to solve it are separated from the parallel implementation. The benefit of such a modular approach is to avoid the time consuming and error prone process of code parallelization that can better be handled by teams with greater experience in computer science disciplines; freeing up time for the researcher to focus on the physical problem. Additionally, the modular design allows for modern and upcoming computational architectures to be fully exploited without large re-writes of the total code base. Other examples of DSLs include Pluto [16], Mint [17], and Devito [18], however unlike OPS these are often restricted to a single computational platform. OPS generates code for MPI, OpenMP, MPI+OpenMP, CUDA, OpenCL and OpenACC, plus MPI versions for multi-GPU clusters.

1.2 Outline of this study

In this paper 3D features of oblique Mach 2 SBLI are investigated for laminar boundary layers with sidewall confinement effects. Disturbances are added to the laminar base flow through blowing/suction applied on the bottom and sidewalls separately. The disturbances trigger a transition to turbulence, with a corresponding reduction in the extent of the shock-induced reverse flow. Section 2 introduces the compressible Navier-Stokes equations in non-dimensional form, and describes the low-dissipative Targeted Essentially non-Oscillatory (TENO) schemes used for the 3D SBLI. TENO schemes potentially represent a significant improvement over the classic Weighted Essentially non-Oscillatory (WENO) schemes. To motivate the choice of TENO schemes for low-dissipative shock capturing, the end of section 2 applies a selection of WENO/TENO schemes to the turbulent 3D Taylor-Green vortex problem. Temporal evolution of integrated enstrophy is shown for each of the shock capturing schemes, compared to a non-dissipative central scheme and a reference result. Section 3 presents a brief overview of OpenSBLI and the OPS code that it generates; full details of the code design and implementation can be found in [13], [14]. Single node runtime performance of the OPS code is given for CPU and GPU. In section 4 the 3D SBLI problem is introduced, with discussion of the computational set-up and sensitivity to grid refinement. Section 5 looks at the structure of the laminar 3D SBLI with sidewall effects, with comparison to a quasi-2D span-periodic configuration. The laminar base flow with sidewalls is then tripped and transitions to turbulence in section 6, to investigate the effect localised disturbances have on the stability of the laminar base flow. Upstream disturbances are amplified by the regions of flow reversal and lead to transition further downstream. A series of hairpin vortices develop as the flow transitions, which merge and form a banded pattern of high velocity streamwise streaks. In the case of disturbances added only to the sidewall, the sidewall separation also showed transition, while the central separation remained laminar.

2 Numerical methods

2.1 Governing equations

The application of OpenSBLI to date has been for aeronautical fluid flow problems, solving the compressible 3D Navier-Stokes equations with finite-difference methods. Despite aeronautics being the main research interest of the authors, OpenSBLI is also capable of generating finite-difference codes for other applications; the core design of OpenSBLI aims to be a general code-generation platform for computational science. In this work OpenSBLI focuses on the compressible Navier-Stokes equations, specifically on supersonic flow physics that requires numerical shock-capturing schemes. OpenSBLI uses numerical indices to distinguish variables, for example the Cartesian coordinate base (x, y, z) and standard velocity components (u, v, w) are taken to be (x_0, x_1, x_2), and (u_0, u_1, u_2) respectively.

The governing compressible Navier-Stokes equations are given in non-dimensional index form with density ρ, pressure p, temperature T, and velocity components u_k as

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_k}\left(\rho u_k\right) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_k}\left(\rho u_i u_k + p\delta_{ik} - \tau_{ik}\right) = 0, \tag{2}$$

$$\frac{\partial}{\partial t}\left(\rho E\right) + \frac{\partial}{\partial x_k}\left(\rho u_k\left(E + \frac{p}{\rho}\right) + q_k - u_i\tau_{ik}\right) = 0, \tag{3}$$

with heat flux q_k and stress tensor τ_{ij} defined as

$$q_k = \frac{-\mu}{\left(\gamma - 1\right) M_\infty^2 Pr Re}\frac{\partial T}{\partial x_k}, \tag{4}$$

$$\tau_{ik} = \frac{\mu}{Re}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} - \frac{2}{3}\frac{\partial u_j}{\partial x_j}\delta_{ik}\right), \tag{5}$$

where Pr, Re and γ are the Prandtl number, Reynolds number and ratio of heat capacities respectively. Unless specifically stated, temperature dependent dynamic viscosity $\mu\left(T\right)$ is given by Sutherland's law

$$\mu\left(T\right) = T^{\frac{3}{2}}\left(\frac{1 + \frac{T_s}{T_\infty}}{T + \frac{T_s}{T_\infty}}\right), \tag{6}$$

with reference and Sutherland temperatures taken to be $T_\infty = 288.0$K and $T_s = 110.4$K. For a freestream Mach number M_∞, pressure and local speed of sound are defined as

$$p = \frac{1}{\gamma M_\infty^2}\rho T, \qquad a = \sqrt{\frac{\gamma p}{\rho}}. \tag{7}$$

2.2 High-order shock capturing schemes

Traditional approaches to shock-capturing include total variation diminishing (TVD), monotonic upstream-centred scheme for conservation laws (MUSCL), and the family of essentially non-oscillatory (ENO) schemes. The past two decades has seen the emergence of more sophisticated methods with superior shock-capturing properties, addressing some of the issues with traditional schemes. The work by [19] gave a comparison of TVD, MUSCL, and ENO, to an early version of a weighted essentially non-oscillatory (WENO) scheme. It was observed that the TVD scheme using a Van-Leer Harmonic limiter added large amounts of numerical diffusion. The MUSCL scheme produced anomalous results, including low frequency oscillations induced downstream of the shock. Better agreement with reference results was obtained via the ENO scheme, but the best shock-resolving performance was obtained by a 5th order WENO formulation.

A more recent review of high-order shock capturing methods [20] was motivated by the need to resolve highly shocked unsteady rocket launch flows. The aim of the review was to replace the existing 2nd order MUSCL scheme within the Launch Ascent and Vehicle Aerodynamics (LAVA) finite difference code, with a modern alternative. Comparison was made between central finite-differencing with artificial dissipation (Central-AD), localized artificial diffusivity (LAD), and weighted essentially non-oscillatory (WENO) schemes. The 6th order Central-AD was the computationally cheapest approach, comparable to the cost of the existing MUSCL scheme while offering better spectral resolution. Despite being an attractive option due to the cheap cost, Central-AD performed poorly compared to the other schemes, and large oscillations were still present across flow discontinuities. LAD schemes work by introducing local grid-dependent artificial diffusion through transport coefficients, smoothing out discontinuities in the flow. These methods obtain good spectral resolution for turbulent flows, and were found to be a credible option for flows containing weak shocks. Their performance over strong shocks however was found to be poor, with significant numerical oscillations observed; both Central-AD and LAD were inferior to the more robust WENO schemes.

Shock capturing methods introduce excessive levels of numerical dissipation, making them a poor choice for resolving small scale structures in transitional and fully turbulent flows. To obtain similar resolution to a non-dissipative scheme, excessively fine grids are required. These large grids are impractical for direct numerical simulation (DNS) and severely limit the scale of the problem that can be tackled. Hybrid methods pair a non-dissipative scheme with shock-capturing applied only in areas of high gradients, reducing the numerical dissipation that is introduced into the flow field. While these methods are widely used, care has to be taken with the choice of shock-sensor, and tuning may be required between different physical problems. Significant efforts have been made to reduce the numerical dissipation of the underlying shock-capturing schemes, such as the improved WENO-Z scheme [21], [22]. Targeted Essentially non-Oscillatory (TENO) schemes [23], [24], aim to improve upon the existing WENO schemes via a discrete stencil selection method and a physically motivated scale separation weighting procedure. TENO schemes are selected to perform the simulations in this paper, owing to the need for resolving a forced transition to turbulence of a laminar separation bubble. The levels of numerical dissipation induced by the scheme are compared to existing WENO schemes for a 3D Taylor-Green vortex turbulent problem at the end of this section. An overview of flux reconstruction schemes are given in the context of a TENO scheme applied to the compressible Navier-Stokes equations (1), where the reconstruction is performed in characteristic space to further reduce numerical oscillations around shocks. Details of the WENO implementation in OpenSBLI is omitted; full details can be found in [14].

2.3 Flux reconstruction

A brief overview of flux reconstruction methods is given in this section, in the context of WENO/TENO schemes. The following is simplified to a 1D advection equation, but it easily extended to the system of equations in (1), where the procedure is applied for each component of the system. Finite-difference WENO/TENO methods have lower computational cost scaling than in finite-volume [25], as the reconstruction can be applied for each dimension of the problem independently. For a simple 1D hyperbolic equation of the form

$$U_t + f(U)_x = 0, \tag{8}$$

the flux term $f(U)_x$ at each discrete grid point x_i can be approximated by computing a flux reconstruction over the half-node locations $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ with grid spacing Δx such that

$$\frac{\partial U}{\partial t} + \frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right) = 0. \tag{9}$$

The full numerical stencil for a WENO/TENO scheme is constructed from a convex combination of lower order candidate stencils, as shown in Figure 1. Non-oscillatory behaviour is obtained by measuring the smoothness in each candidate stencil, and assigning it a weighting ω_r in the final reconstruction. Fluxes at the half-node locations $f_{i+\frac{1}{2}}$ and $f_{i-\frac{1}{2}}$ are reconstructed as a weighted sum of essentially non-oscillatory (ENO) interpolations over a set of r candidate stencils

$$\hat{f}_{i+\frac{1}{2}} = \sum_{r=0}^{K-3} \omega_r \hat{f}^{(r)}_{i+\frac{1}{2}}, \tag{10}$$

where $\hat{f}^{(r)}_{i+\frac{1}{2}}$ are the classic ENO interpolations given in [25], and ω_r are the non-linear weights satisfying the conditions

$$\omega_r \geq 0, \quad \text{and} \quad \sum_{r=0}^{K-3} \omega_r = 1. \tag{11}$$

The formal accuracy of the scheme K results in three and four candidate stencils for the fifth and sixth order TENO schemes respectively. Flux splitting is applied by summing upwind/downwind biased reconstructions

$\hat{f} = \hat{f}^{+} + \hat{f}^{-}$, using the well known local Lax-Friedrichs splitting method

$$f^{\pm}(U) = \frac{1}{2}\left(f(U) \pm \alpha U\right), \tag{12}$$

for a locally evaluated wave-speed α over the full numerical stencil. The negative flux contribution is obtained by reflecting the stencils in Figure 1 about the half-node reconstruction point $x_{i+\frac{1}{2}}$. The next section describes how the non-linear weights ω_r are calculated, and the differences TENO has to conventional WENO reconstructions.

2.4 Targeted Essentially non-Oscillatory (TENO) schemes

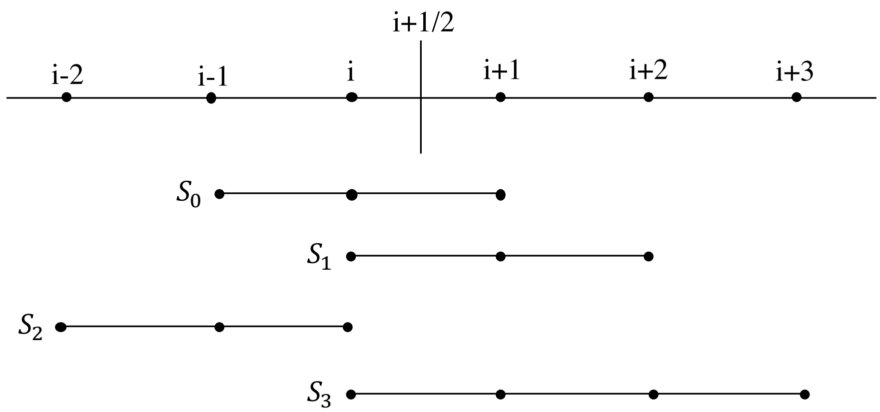


Figure 1: Staggered candidate stencils for TENO schemes, adapted from [23].

TENO schemes differ from WENO in three fundamental ways: the staggered ordering of candidate stencils as in Figure 1, the complete removal of candidate stencils deemed to be non-smooth via a discrete cut-off function, and the modified non-linear weights, optimized for low numerical dissipation. The stencil staggering is claimed to reduce the number of candidate stencils that are crossed by a discontinuity [23], helping to enforce the underlying principles of the base ENO method. In the standard WENO formulation each candidate stencil is given a non-linear weighting, with candidate stencils crossing discontinuities given a low, albeit non-zero, weight. TENO schemes abandon this approach in favour of a discrete cut-off function that discards candidate stencils completely from the flux reconstruction if they are deemed to be sufficiently non-smooth. Candidate stencils that are considered smooth are included in the reconstruction with their ideal linear weight to further reduce numerical dissipation. In reference to Figure 1, the 5th order reconstruction uses candidate stencils $\{S_0, S_1, S_2\}$, and the 6th order scheme uses $\{S_0, S_1, S_2, S_3\}$. The non-linear weights of [25] are reformulated with ideal weights d_r for a scheme of order K to be

$$\omega_r = \frac{d_r \delta_r}{\sum_{r=0}^{K-3} d_r \delta_r}, \tag{13}$$

where δ_r is a discrete cut-off function of the form

$$\delta_r = \begin{cases} 0 & \text{if} \quad \chi_r < C_T \\ 1 & \text{otherwise} \end{cases}$$

for a tunable cut-off parameter C_T. The smoothness measures χ_r are the same as the weight normalization process in WENO

$$\chi_r = \frac{\gamma_r}{\sum_{r=0}^{K-3} \gamma_r}, \tag{14}$$

comprised of the WENO-Z [21] inspired form of non-linear TENO weights [23] given by

$$\gamma_r = \left(C + \frac{\tau_K}{\beta_r + \epsilon} \right)^q, \quad r = 0, \ldots, K-3. \tag{15}$$

Polynomial smoothness indicators β_r are unchanged from the standard Jiang-Shu formulation [26], and are given explicitly for the TENO stencils of Figure 1 in [23]; the small parameter $\epsilon \sim 10^{-40}$ is used to avoid division by zero. A global smoothness indicator τ_K measures smoothness over the full numerical stencil, and is given for 5th and 6th order TENO schemes as

$$\tau_5 = |\beta_0 - \beta_2|, \tag{16}$$

$$\tau_6 = |\beta_3 - \frac{1}{6}(\beta_0 + \beta_2 + 4\beta_1)|. \tag{17}$$

One feature of the TENO schemes is the optimization of the parameters C and q within the non-linear weights, and the tunable cut-off parameter C_T found in the discrete weight selection function (2.4). The parameters C and q control the levels of dissipation invoked by the non-linear weight, and are based on physically-motivated scale-separation mechanisms [23]. TENO schemes take the values of $q = 6$ and $C = 1$, which was shown to significantly enhance the discontinuity detection capability of the scheme [23]. C_T meanwhile is the parameter that determines whether a given candidate stencil is rejected or contributes to the flux reconstruction. Lower values of C_T are suitable for compressible turbulence simulations where minimal numerical dissipation is desirable, but this comes at the cost of increased spurious oscillations around shock-waves. In the work of [23] an optimization process was performed, with the recommended optimal values of C_T for 5th and 6th order TENO schemes stated to be 1×10^{-5} and 1×10^{-7} respectively.

2.5 Characteristic reconstruction for systems of equations

Multi-dimensional problems are solved by applying the TENO reconstruction procedure component by component, and then in each spatial dimension independently. While the method can be applied to either a primitive or conservative formulation of the governing equations (1), reduced oscillations around discontinuities can be obtained if the equations are first transformed into characteristic space [20]. For solution and flux vectors U and $f(U)$ of size m for the system of equations (1), there exists an eigensystem to decompose the equations into characteristic space. For a set of m eigenvalues

$$\lambda_1(U) \leq \cdots \leq \lambda_m(U), \tag{18}$$

with a set of independent eigenvectors $r(U)$ that form the columns of the right eigenvector matrix

$$R(U) = (r_1(U), \ldots, r_m(U)), \tag{19}$$

the Jacobian of the system can be diagonalised such that

$$R^{-1}(U) f'(U) R(U) = \Lambda(U). \tag{20}$$

To perform a reconstruction on the half-node location $x_{i+\frac{1}{2}}$, the matrices $R(U)$, $R^{-1}(U)$ must be evaluated at an averaged state using either simple or Roe-averaging; for this study Roe-averaging is used throughout. Each component of the split form of the flux (12) can be transformed into characteristic space by multiplying it by the averaged left eigenvector matrix $R^{-1}(U_{i+\frac{1}{2}})$. For characteristic systems the wave-speed α in (12) is taken for each component to be the maximum eigenvalue (18), evaluated over the full numerical stencil in Figure 1. Having formed the full flux reconstruction in characteristic space, the flux is transformed back to physical space by multiplying it by the averaged right eigenvector matrix $R(U_{i+\frac{1}{2}})$. The final step is to build the difference approximation as in (9)

$$\frac{1}{\Delta x} \left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right), \tag{21}$$

where $\hat{f}_{i-\frac{1}{2}}$ would have been evaluated from the neighbouring grid point, and the flux contains the summed upwind/downwind contributions

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}^{+}_{i+\frac{1}{2}} + \hat{f}^{-}_{i+\frac{1}{2}}. \tag{22}$$

2.6 Viscous terms and time advancement

Viscous terms of the compressible Navier-Stokes equations defined in (1) do not require a special treatment with WENO/TENO, and can simply be computed with standard central differencing. To improve numerical stability the Laplacian terms are expanded; one-sided derivatives are used as a boundary closure at all domain boundaries aside from periodic. The one-sided derivative formulation is that of [27], with the derivative evaluation at the first four points at domain boundaries replaced by alternative weightings. The central differencing inside the domain is kept at fourth order in this study to match the fourth order of the one-sided boundary treatment; OpenSBLI is capable of generating higher order central schemes as required.

Large-scale direct numerical simulation (DNS) of the compressible Navier-Stokes equations has considerable memory requirements, making low-storage time-advancement schemes an attractive option. OpenSBLI currently has two low-storage explicit time-stepping schemes available, a standard 3rd order Runge-Kutta scheme described in [13], and a strong stability preserving (SSP) version designed for flows containing discontinuities. The SSP version is a 3-stage 3rd order Runge-Kutta scheme in the low-storage form proposed by [28], requiring only two storage registers per equation. For an m-stage scheme, time advancement of the solution vector u from level u^n to u^{n+1} is performed at stage $i = 1, \ldots, m$ such that

$$du^{(i)} = A_i du^{(i-1)} + \Delta t R\left(u^{(i-1)}\right), \tag{23}$$

$$u^{(i)} = u^{(i-1)} + B_i du^{(i)}, \tag{24}$$

$$u^{n+1} = u^{(m)}, \tag{25}$$

for a constant time-step Δt, initial conditions $u^{(0)} = u^n$ and $du^{(0)} = 0$, and residual $R(u^{(i-1)})$. The constants A_i, B_i are taken to enforce the SSP condition as introduced in [29].

2.7 Scheme comparison: 3D Taylor-Green vortex

Conventional shock-capturing schemes introduce excessive levels of numerical dissipation, making them unsuitable for resolving small-scale turbulent structures. To demonstrate the effect numerical dissipation has on the accuracy of a solution, the 3D Taylor-Green vortex problem was simulated for a selection of shock-capturing schemes present in OpenSBLI. Results were compared to a non-dissipative 4th order central difference scheme, and high-resolution spectral reference data [30]. Problem specifications are taken from [31], solving the compressible Navier-Stokes equations on a domain of size $0 \leq x_0 \leq 2\pi L$, $0 \leq x_1 \leq 2\pi L$, and $0 \leq x_2 \leq 2\pi L$. Periodic boundary conditions were applied in all directions. The domain was initialised as

$$u_0(x_0, x_1, x_2, t=0) = \sin\left(\frac{x_0}{L}\right)\cos\left(\frac{x_1}{L}\right)\cos\left(\frac{x_2}{L}\right), \tag{26}$$

$$u_1(x_0, x_1, x_2, t=0) = -\cos\left(\frac{x_0}{L}\right)\sin\left(\frac{x_1}{L}\right)\cos\left(\frac{x_2}{L}\right), \tag{27}$$

$$u_2(x_0, x_1, x_2, t=0) = 0, \tag{28}$$

$$p(x_0, x_1, x_2, t=0) = \frac{1}{\gamma \mathrm{M}^2} + \frac{1}{16}\left(\cos\left(\frac{2x_0}{L}\right) + \cos\left(\frac{2x_1}{L}\right)\right)\left(2 + \cos\left(\frac{2x_2}{L}\right)\right). \tag{29}$$

Reynolds, Prandtl and Mach numbers were set to $\mathrm{Re} = 1600$, $\mathrm{Pr} = 0.71$, and $\mathrm{M} = 0.1$ respectively; the ratio of heat capacities was taken to be $\gamma = 1.4$. The reference quantities L, u_{ref} and ρ_{ref} were set to 1.0, and the reference temperature T_{ref} was evaluated using the equation of state (7).

The simulation was performed on a 256^3 uniformly spaced grid, with a non-dimensional time-step $\Delta t = 8.4625\times 10^{-4}$ [31]. The integral of enstrophy is evaluated during the simulation as in [31] such that

$$\varepsilon = \frac{1}{\rho_{\text{ref}}\Omega} \int_{\Omega} \frac{1}{2}\rho \left(\epsilon_{ijk} \frac{\partial u_k}{\partial x_j} \right)^2 \, \mathrm{d}\Omega, \tag{30}$$

where Ω is the entire volume of the domain and ϵ_{ijk} is the Levi-Civita function.

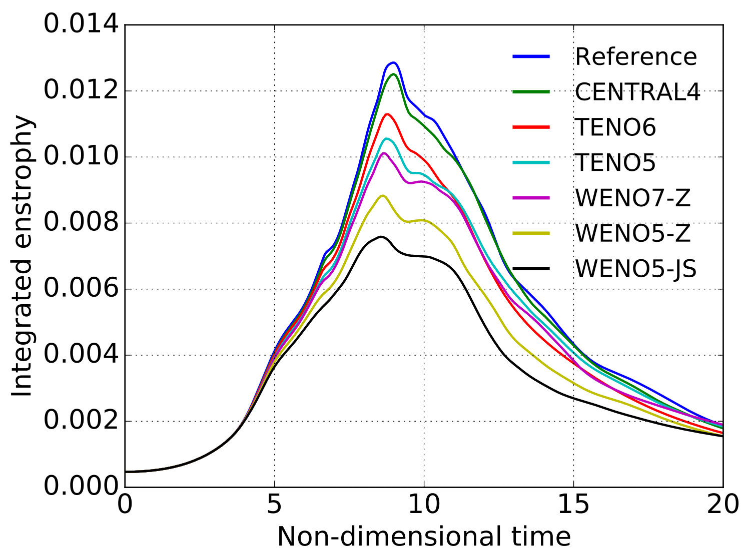


Figure 2: Temporal evolution of enstrophy for the 3D Taylor-Green vortex problem on 256^3 grids. The reference solution is the 512^3 spectral result from [30].

Figure 2 shows the resolving ability for a range of WENO/TENO schemes compared to the non-dissipative central scheme. The standard 5th order WENO-JS scheme of [25] severely underestimates the enstrophy curve, with the peak enstrophy reaching only about half of the reference solution. The WENO-Z schemes of [21] offer a substantial improvement over WENO-JS of equal order; the different formulation of the non-linear weights gives reduced numerical dissipation for a negligible increase in computational cost [22]. Both the 5th and 6th order TENO schemes were less dissipative than even the 7th order WENO-Z scheme, with the 6th order TENO scheme only underestimating the enstrophy peak by around 10% compared to the non-dissipative central scheme. The computational cost of the TENO schemes is $\sim 15\%$ greater than the equivalent order WENO, owing to the evaluation of discrete cut-off functions (2.4) for stencil selection.

While the performance of the TENO schemes for a turbulent test case is much improved from WENO formulations, they still lag behind the non-dissipative scheme. Further reduction in the induced numerical dissipation could be achieved via the use of a Roe-flux instead of local Lax-Friedrichs flux splitting, or with TENO schemes of order higher than 6 [24]. Hybrid methods, blending a non-dissipative scheme to a shock-capturing scheme, are still preferable for the reduction of numerical dissipation for fully turbulent problems. For simplicity the 6th order TENO scheme is selected to perform the simulations in this paper, as it offers sharp shock-capturing ability and greatly reduced numerical dissipation compared to more established schemes.

3 Code generation & domain specific languages

In this section a brief overview of the Python-based source-code generator OpenSBLI [13], [14] is given, demonstrating how code generation techniques can be paired with a domain-specific language to solve scientific problems. Users of OpenSBLI provide a high-level Python script containing the equations they wish to solve, using the numerical schemes and boundary conditions made available to them. Section 3.1 shows a cut-down version of one of these problem scripts, showing the steps required to solve the continuity equation.

Computations to be performed are segregated into computational kernels. An example of an OPS kernel is given in section 3.2 and the corresponding parallel execution call is shown in section 3.3. Finally, the ability of OPS to generate code compatible with different architectures in demonstrated in section 3.4, with single node performance for a 3D SBLI case shown for CPU and GPU systems.

3.1 OpenSBLI problem file

```
1  mass = "Eq(Der(rho,t), - Conservative(rhou_j,x_j,**{scheme:Teno}))"
2  Simulation_eqn = SimulationEquations(); ndim = 3 # Instantiate the main equation class
3  Simulation_eqn.add_equations(EinsteinEquation().expand(mass, ndim, coordinate_symbol="x",
       substitutions=[], constants=[])) # Parse and expand the equation over the specified dimensions
4  LLF = LLFTeno(teno_order=6, RoeAverage([0, 1])) # Select a 6th order TENO scheme with Roe averaging
5  RK = RungeKuttaSSP(3) # Select a 3rd order SSP RK time-stepping scheme
6  block = SimulationBlock(ndim, block_number=0) # Create a block
7  block.set_equations([Simulation_eqn]) # Set the equations to be solved
8  block.set_discretisation_schemes({LLF.name: LLF, RK.name: RK}) # Set numerical schemes to be used
9  block.discretise() # Discretise the equations on the block
10 alg = TraditionalAlgorithmRK(block) # Create a Runge-Kutta solution algorithm
11 OPSC(alg) # Generate and write out the OPS C code
```

Listing 1: Simplified OpenSBLI Python problem file to generate a code to solve the continuity equation.

A cut-down version of an OpenSBLI problem file is shown in the code snippet above, containing the steps required to generate a code to solve the continuity equation (1). In line 1 the equation is written as a string in index form, to be expanded over the index j. The SymPy equation class *Eq* is used with time and spatial derivatives on the left and right hand sides of the equation respectively. A spatial conservative derivative is specified with a TENO scheme; changing the scheme here would modify how this derivative is discretised. Lines 2-3 parse and expand the equation over the number of user-selected dimensions, before adding them to the *SimulationEquation* class that stores the equations to be advanced in time. Line 4 instantiates a characteristic local Lax-Friedrichs 6th order TENO scheme with Roe averaging, while third order Runge-Kutta time-stepping is selected in line 5. A single *SimulationBlock* is created for this simulation in line 6, to which the equations to solve and numerical schemes are set on in lines 7-8. At this point the user would also instantiate classes for boundary conditions (in each direction), initial conditions and file I/O options, which are each set on the block in the same way.

The block now contains everything required to solve the problem. The call to the discretisation method in line 9 begins the discretisation of the governing equations for the selected schemes. During code generation each significant computation is stored as a computational kernel object, which contains the required calculation, grid ranges, data access and input/output arrays. In line 10 an algorithm is generated; the algorithm class loops over all of the kernels created by the discretisation, and orders them at the appropriate position for the algorithm. Line 11 calls the OpenSBLI OPS C code writer, which finds each of the components of the algorithm and writes them to a file as OPS compliant C code. The final code is then passed to the OPS code translator, which generates parallel versions of the supplied sequential source code.

3.2 OPS Computational kernel

```
void opensbliblock00Kernel035(const double *u0_B0, double *wk1_B0){ // Kernel arguments
wk1_B0[OPS_ACC1(0,0,0)] = inv_1*(-rc9*u0_B0[OPS_ACC0(2,0,0)] - rc8*u0_B0[OPS_ACC0(-1,0,0)] +
    (rc8)*u0_B0[OPS_ACC0(1,0,0)] + (rc9)*u0_B0[OPS_ACC0(-2,0,0)]);} // Computing a derivative
```

Listing 2: Example OPS kernel code generated by OpenSBLI. The kernel computes a 4th order central finite-difference approximation of the variable u_0.

The code snippet above is an example of the OPS code generated by OpenSBLI for a sample calculation. The simplified kernel calculates a 4th order central derivative of the velocity component u_0 with respect to x_0. A temporary work array *wk1* has been assigned to this kernel to store the result of the computation. For each of the arrays there is an OPS_ACCX(0,0) argument, where X is an integer referring to the position of that array in the function arguments. The quantities in parentheses are the relative grid indexing, with $(0, 0, 0)$ referring to the current grid point the kernel is being evaluated on. In OPS all computations are specified in this way, with the user providing the computation to be performed relative to the current grid point. The *inv_1* and *rcXX* quantities are rational constants. OpenSBLI substitutes place holders to be globally defined, for all of the rational constants found in the equations during code generation. Rational quantities in the output code contain these placeholder variables and the numerical values are only computed once when the program is initialized.

3.3 Corresponding OPS function call

```
int iteration_range_35[] = {0, block0np0, 0, block0np1, 0, block0np2}; // Grid range to compute
ops_par_loop(opensbliblock00Kernel035, "Viscous CD u0_B0 x0", opensbliblock00, 3,
    iteration_range_35, // Parallel loop to perform a central derivative
ops_arg_dat(u0_B0, 1, stencil_0_08, "double", OPS_READ), // Input array and stencil access
ops_arg_dat(wk1_B0, 1, stencil_0_00, "double", OPS_WRITE)); // Output array
```

Listing 3: Example OPS parallel loop call, generated automatically by OpenSBLI. The loop executes the kernel code over the entire 3D grid range.

The kernel code in the previous section specifies the computation to be performed, but does not contain information relating to its execution over a grid. This is instead decoupled into the main program file, where the calls to that kernel will be placed in the position required by the numerical algorithm. In the code above, the first line defines the iteration range to perform the calculation; in this case the range is taken to be the entire 3D grid. A call is made to an *ops_par_loop*, which is a parallel region in which OPS has control over how the computation is decomposed and executed. Each of these parallel loops within the base source code will be parsed by the OPS translator, and converted into different versions of parallel code ready for the user to compile. OPS parallel loops require the array input/output arguments and their read/write status, desired floating point precision, and a stencil. Stencils are lists of integers to tell OPS the data access required to perform the computation, relative to each grid point; in this case the input stencil requires access from $i-2$ to $i+2$ in the x_0 direction. The role of OpenSBLI is to generate these kernels, extract their data dependencies, stencil access and iteration ranges, and write out strings compliant with the OPS language. While OPS code can be written manually by hand, the added layer of code generation removes much of the tedious work and reduces errors.

3.4 OPS architecture performance comparison

Architecture/compiler	**Runtime (s)**	**Speed-up**
Intel Skylake (40 cores @ 2 GHz, 40 MPI, Intel 17.0 -O3 -fp-model fast)	174.1	1.0
NVIDIA Pascal 16GB P100 (CUDA 8.0, nvcc -O3)	54.5	3.2
NVIDIA Volta 16GB V100 (CUDA 9.0, nvcc -O3)	35.2	4.9

Table 1: OPS single node runtime comparison on different architectures for 100 iterations. The time for the CPU node (Intel Skylake Xeon Gold 6138) with 40 MPI processes is taken as the baseline.

A runtime comparison was performed to assess OPS single node performance for the 3D laminar SBLI case presented in the next section. A grid size of $(Nx_0, Nx_1, Nx_2) = (500, 170, 170)$ was selected to fit on a single GPU, with the time for 100 iterations of the main iteration loop recorded. The relative performance between a GPU and a conventional CPU node is shown in Table 1. A baseline time of 174.1 seconds is recorded for the Intel Skylake node, with one MPI process per core and the AVX-512 instruction set applied. Stencil-based codes on structured meshes with explicit time-stepping are known to be well suited to GPU architectures, which is consistent with the timings obtained for this problem. A single NVIDIA P100 Pascal

GPU was 3.2 times faster than the CPU node with a time of 54.5 seconds. The latest NVIDIA V100 Volta GPU improved upon this, with the single GPU being equivalent to slightly under 200 CPU cores.

Aside from the difficulty of porting scientific codes to run on GPUs, the main drawback of GPU platforms is the limited available memory capacity. While the compute capability per card is impressive, the difficulty is in providing each card with enough work using the limited memory available. To reduce the memory requirements of the schemes, the implementation of WENO/TENO schemes in OpenSBLI attempts to minimize the number of global arrays used for the reconstruction. Further optimizations of this kind are possible within a code-generation framework. An initial re-factoring of the algorithms was reported in [32], for central differencing in OpenSBLI. By using the OPS library the code is not restricted to a single computational architecture and the performance can be assessed on a case by case basis for different simulations. As the implementation of numerical algorithms in OpenSBLI is independent of the parallel implementation, the code is better placed to adapt to future changes in the high performance computing ecosystem.

4 3D SBLI problem specification

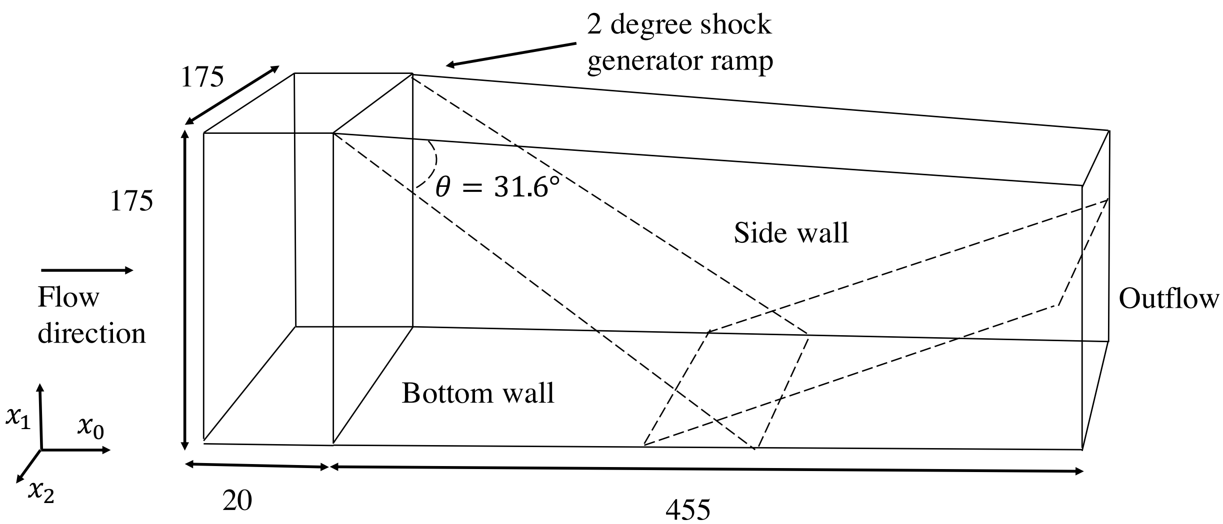


Figure 3: Schematic of the computational domain in non-dimensional units.

Three-dimensional shockwave/boundary-layer interactions are simulated for a Mach 2 inlet to a rectangular duct; a schematic of the computational domain is shown in Figure 3. An incident oblique shockwave is generated by an internal ramp on the upper surface of the domain angled at 2°, giving a shock angle of 31.6°; the shockwave impinges on a laminar boundary-layer and is reflected. In addition to the main interaction region, the presence of sidewalls on either side of the span causes a swept shock interaction with the sidewall boundary-layers. The dimensions of the domain are $(L_{x_0}, L_{x_1}, L_{x_2}) = (475, 175, 175)$, corresponding to a one-to-one aspect ratio of the cross-sectional area. The height of the domain was selected to ensure that reflections from the upper boundary do not impinge on the lower wall boundary-layer.

The domain is initialised with a similarity solution to the compressible boundary-layer equations [33]. Boundary layers are initialised on the bottom wall and both of the sidewalls, with a blending of the two profiles applied in the corner region where two physical walls meet. The blending is achieved by fitting high order polynomials to the profiles generated by the similarity solution, which are multiplied together in the corner region. The wall normal velocity component from the similarity solution is linearly reduced with increasing distance from the sidewall, to obtain a zero u_2 velocity component at the mid-point of the span.

Most numerical work on SBLI to date has been restricted to 2D and span-periodic 3D configurations, for which the incident shock can easily be imposed as a Dirichlet boundary condition containing exact shock-jump conditions. However, applying this method in the presence of sidewalls proved to be problematic at the interface between the upper shock conditions and the viscous sidewalls. Attempting to match supersonic shock conditions on the upper boundary with a subsonic sidewall boundary-layer generated substantial numerical artefacts from the discontinuity. To remedy this problem, it was necessary to use a curvilinear mesh to physically deflect the flow with a ramp. The ramp is taken to be an inviscid slip wall, acting as a method to generate the incident shock and allow for the main and swept shock/boundary-layer interactions to be investigated. A viscous no-slip ramp was not feasible for this study since compression waves emitted from the start of the separation region on the bottom wall would cause boundary-layer separation on the ramp. Compression waves from the ramp separation would cause a further separation on the bottom wall which would then merge with the primary separation.

The bottom wall and both sidewall boundaries are set to viscous no-slip isothermal conditions, for the constant wall temperature obtained from the similarity solution [33]. The inlet plane is treated with a pressure-based extrapolation method dependent on the local speed of sound. At the outlet a zeroth-order extrapolation of all flow variables is applied. The inviscid wall ramp is located at $x_0 = 20$ with a deflection angle of $2°$; upstream of the ramp, freestream and sidewall boundary-layer conditions are maintained with constant freestream pressure. The initial shock impinges on the boundary-layer at $Re_x = 3.7 \times 10^5$, and the Reynolds number based on inlet displacement thickness is 950. Dynamic viscosity is used with Sutherland's law (6), for Sutherland and reference temperatures of $110.4K$ and $288.0K$ respectively. The Prandtl number and ratio of heat capacities are 0.72 and 1.4, with a non-dimensional wall temperature of $T_w = 1.6762$ (5 s.f.).

4.1 Computational set-up and grid sensitivity

Grid	Total points	$\mathbf{N_{x_0} \times N_{x_1} \times N_{x_2}}$	Min $\mathbf{\Delta x_0, \Delta x_1, \Delta x_2}$	Max $\mathbf{\Delta x_0, \Delta x_1, \Delta x_2}$
Coarse	26.6×10^6	$525 \times 225 \times 225$	0.90, 0.24, 0.24	0.90, 1.30, 1.30
Medium	65.1×10^6	$700 \times 305 \times 305$	0.68, 0.17, 0.17	0.68, 0.95, 0.95
Fine	123.0×10^6	$875 \times 375 \times 375$	0.54, 0.14, 0.14	0.54, 0.78, 0.78

Table 2: Computational grid parameters for three resolutions. The reported spacings are in non-dimensional units at the inlet plane.

Table 2 shows the three grids used to determine grid sensitivity, with the minimum and maximum spacings in each direction taken at the inlet of the domain. In each case x_0 was uniformly distributed, x_1 was clustered symmetrically for higher grid density on the bottom wall and around the shock-generator, and x_2 was clustered symmetrically towards both sidewalls. For direction based stretch factors $s_1 = s_2 = 1.5$, uniformly distributed points in the interval $\xi = [0, 1]$ and lengths Lx_i the grid stretching functions were

$$x_0 = L_{x_0}\xi, \tag{31}$$

$$x_1 = \frac{Lx_1}{2}\left(1 - \frac{\tanh\left(s_1(1-2\xi)\right)}{\tanh\left(s_1\right)}\right), \tag{32}$$

$$x_2 = \frac{Lx_2}{2}\left(1 - \frac{\tanh\left(s_2(1-2\xi)\right)}{\tanh\left(s_2\right)}\right). \tag{33}$$

All simulations were performed with a 6th order TENO scheme for spatial discretisation, with 3rd order Runge-Kutta explicit time-stepping and a non-dimensional time-step of $\Delta t = 0.025$. For the laminar base flow in the following section, the solution was advanced to a non-dimensional time of $t = 1 \times 10^4$. Convergence was ascertained by monitoring the growth of the skin-friction on the bottom wall, taken along the centreline of the span. The non-dimensional time of $t = 1 \times 10^4$ corresponds to $\sim$ 21 flow through times of the computational domain. Figure 6 (a) shows the sensitivity of the laminar solution to grid refinement. The coarse grid shows numerical defects in the skin-friction at the inlet, and slightly underestimates the length of the region of reverse flow. The medium grid is very close to the finest result, but exhibits some small oscillations in the skin-friction near the outlet. The finest grid is used for all results in the subsequent sections unless otherwise stated.

5 3D Shockwave/boundary-layer interaction with sidewall effects

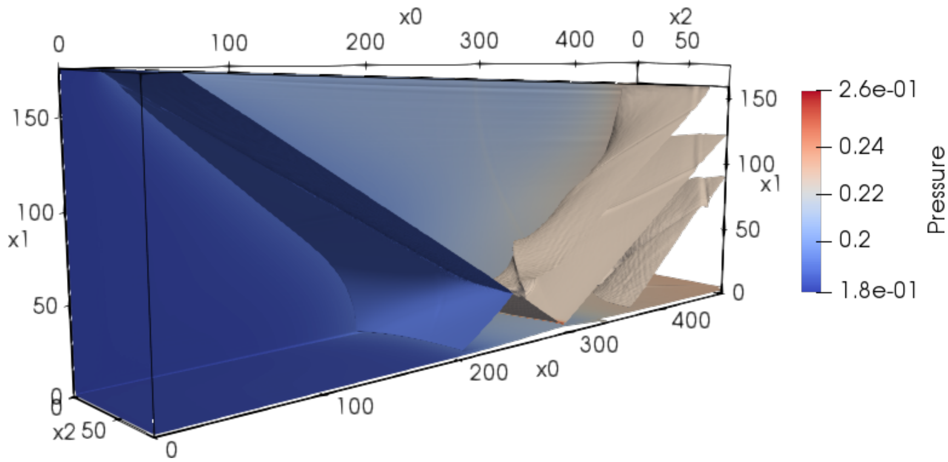


Figure 4: Density contours coloured by pressure; half of the channel for the laminar 3D SBLI is shown.

In this section the results for the laminar 3D SBLI with sidewall effects are presented, simulated on the $(Nx_0, Nx_1, Nx_2) = (875 \times 375 \times 375)$ finest grid from the previous section. Figure 4 shows density contours coloured by pressure for the final flow field at $t = 1 \times 10^4$; half of the spanwise length is plotted as the laminar problem is symmetric about the centreline. The flow is turned by the ramp located at $x_0 = 20$, causing a swept oblique shockwave that reflects off the bottom laminar boundary layer. The adverse pressure gradient of the incident shock thickens the bottom and side wall boundary layers, causing large regions of reverse flow ($u_0 \leq 0$) as seen in Figure 5. The shape of the incident shock deviates considerably as a result of the sidewall, a feature that is reduced closer to the centre of the span. The swept interaction of the incident shock with the sidewall boundary layer causes a visible curving that is not seen in span-periodic configurations. Compression waves emitted as the boundary-layer thickens near the centreline are noticeably further downstream compared to those emanating from the corner separation. Three main features are identified in the vicinity of the bottom boundary layer: compression waves from the start of the central separation, and expansion fans at the top of the central separation and at the reattachment point. All three features leave the domain through the outlet, and do not cause secondary impingement on the bottom wall boundary layer.

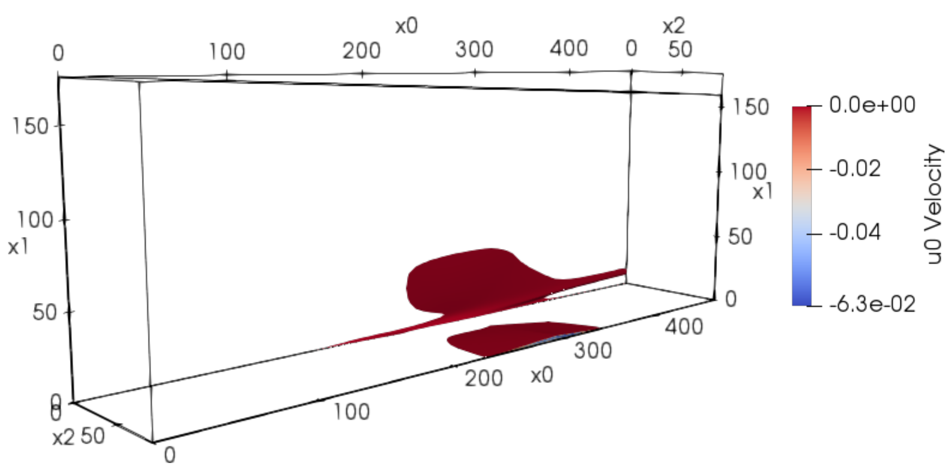
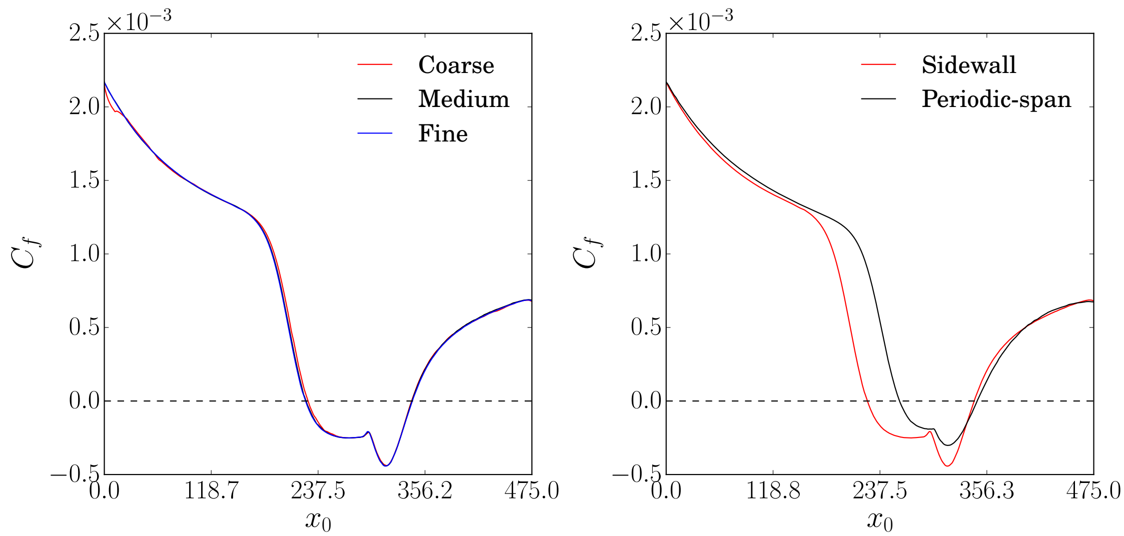


Figure 5: Regions of centre, corner, and sidewall reverse flow $u_0 \leq 0$; half of the channel for the laminar 3D SBLI is shown.



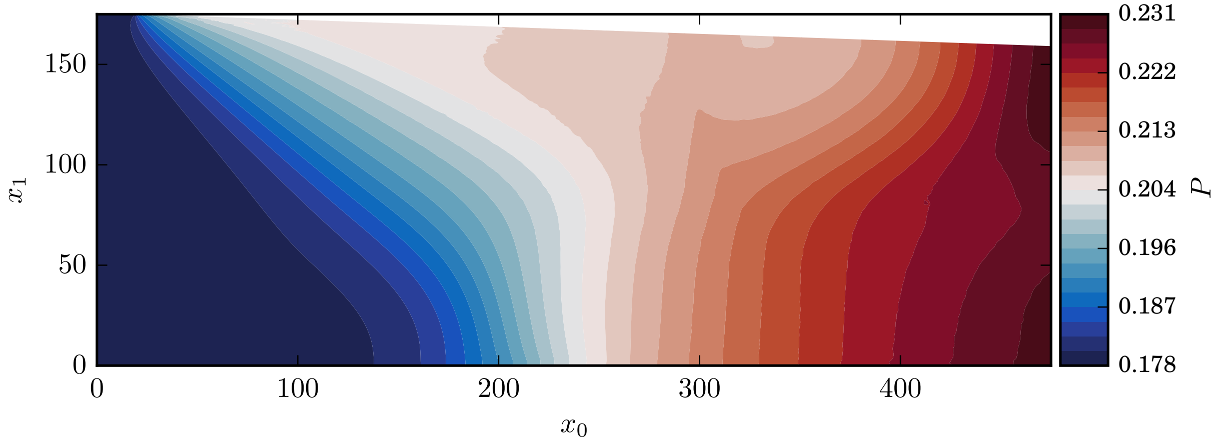
(a) Grid sensitivity of centreline skin-friction. (b) Effect of sidewall confinement on separation.

Figure 6: Streamwise skin-friction distribution along the bottom wall, evaluated at the centre of the span.

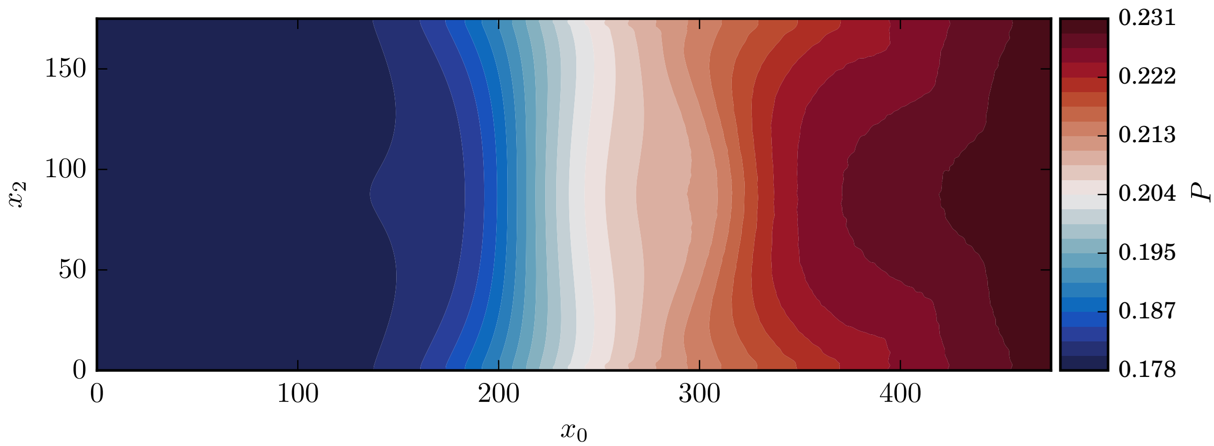
A span-periodic configuration with identical parameters was also simulated to highlight the effect that sidewalls have on the central separation region. The grid was modified only in the spanwise direction, with uniform spacing along the span and a resolution of $(Nx_0, Nx_1, Nx_2) = (875 \times 375 \times 225)$. Figure 6 (b) shows the effect sidewall confinement has on the centreline skin friction. In both cases the expected form of the skin-friction for a laminar separation bubble is observed, with reversed flow seen under the dashed black line $C_f = 0$. The asymmetric shape of the separation bubble with two distinct troughs in Figure 6 (b) relates to the pressure rises as the boundary layer separates, and then reattaches further downstream. Confinement effects greatly increase the length and magnitude of the separation bubble, since the swept sidewall-shock interaction strengthens the incident shockwave significantly. At this one-to-one aspect ratio of duct width to height, the failure of the span-periodic approximation to predict the extent of centreline flow separation is clear to see. A similar picture is seen in the side and bottom wall pressure contours of Figures 7 (a) and (b) respectively. The swept interaction between the sidewall boundary layer and the incident shock causes an initial jump in pressure far upstream of the 2D impingement location, corresponding to a region of separated flow in the corner region. Figure 7 (b) shows the pressure distribution along the bottom wall, consistent with the picture in Figure 4. At the start of the separation bubble in the middle of the span the flow tends towards quasi-2D results and pressure contours increase perpendicular to the streamwise direction. In contrast, at each of the corner regions there is a clear curving of the flow, where the strength of the sidewall effect is at its greatest.

Wall-normal components of skin-friction are plotted for the side and bottom walls in Figures 8 (a) and (b) respectively. A large region of separated flow is observed on the sidewall in Figure 8 (a), beginning at the corner separation around $\sim x_0 = 150$. The zero crossing of the skin-friction is displayed as the dashed white line, demonstrating that the sidewall separation extends almost a third of the way up the height of the sidewall. A long thin streak of reverse flow is seen propagating all the way to the outlet. Further work with a longer computational domain is needed, to identify whether this is a numerical artefact from the extrapolation outlet boundary condition or not. Some numerical noise is observed at the start of the ramp on the sidewall; this artefact was more pronounced for the two coarser grids and may be reduced with further grid refinement.

Figure 8 shows the shape of the separation bubble on the bottom wall, once again encircled by the dashed white zero-crossing line. The separation bubble takes an asymmetric oblate circular shape, curved at the downstream side of reattachment and essentially perpendicular to the freestream near the onset of separation. Two central core regions of highest magnitude consistent with the line plot of Figure 6 can be seen, with the interaction at its strongest directly on the centreline. Two elongated thin strips of corner



(a) Instantaneous pressure contours on the sidewall $x_2 = 0$.

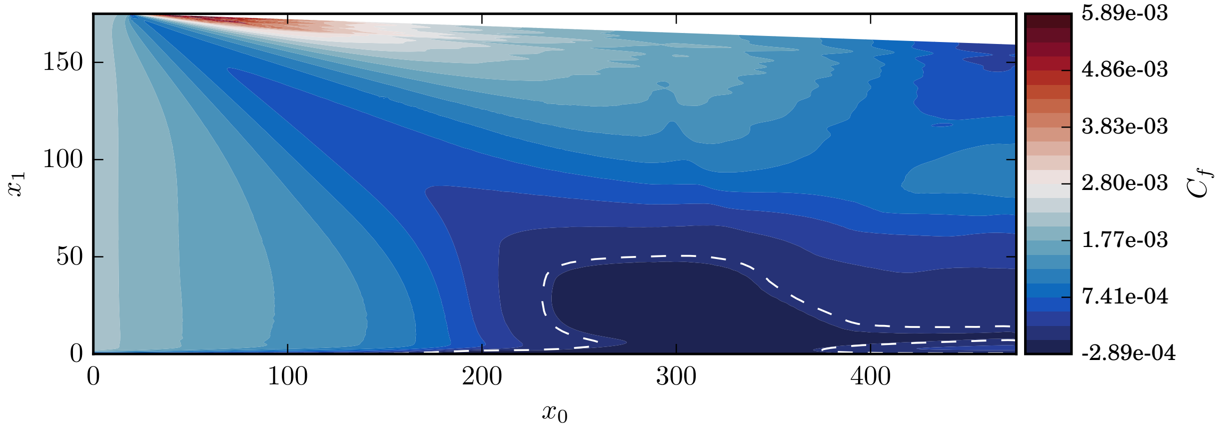


(b) Instantaneous pressure contours on the bottom wall $x_1 = 0$.

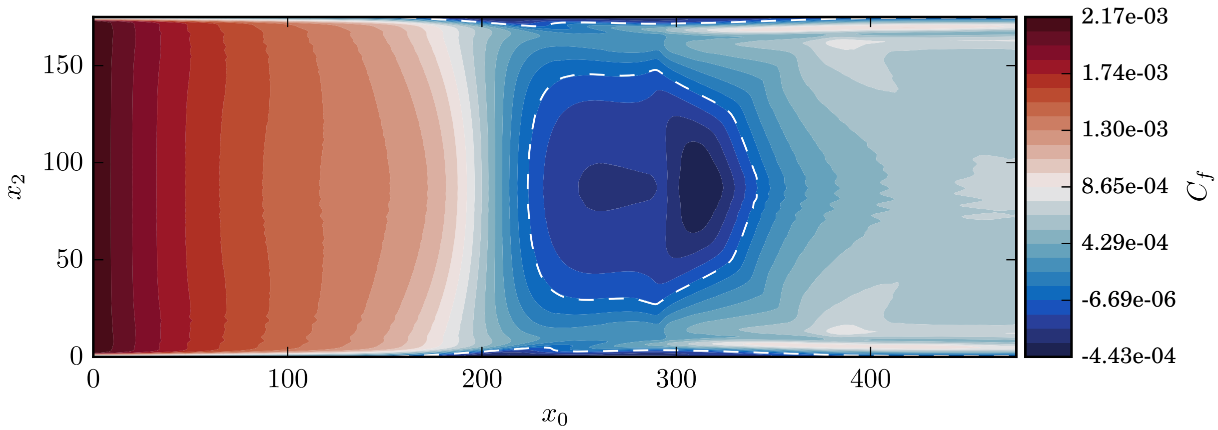
Figure 7: Side (a) and bottom wall (b) pressure distributions for the 3D laminar SBLI with sidewall effects.

separation are observed in the near sidewall region, with a buffer region of attached flow between them and the central separation. Both the main and sidewall separations are substantially larger than those observed in the turbulent study of [7], owing to the relative ease at which separation of a laminar boundary-layer can occur.

The laminar base flow demonstrates the importance of modelling confinement effects when dealing with shockwave/boundary layer interactions for internal flows. Regions of corner and sidewall separation not captured in quasi-2D simulations have a drastic impact on the main interaction. To further highlight the inadequacy of span-periodic simulations and their inability to replicate wind tunnel configurations, the three-dimensionality of the interaction for a confined flow is shown in the velocity streamlines of Figure 9. Figure 9 (a) displays (u_0, u_2) velocity streamlines within the central separation bubble, evaluated one point above the bottom wall and coloured by pressure contours. Reverse flow can be seen in the corner regions at $x_2 = 0, 175$, and high velocity streaks in the attached region that merge into the central separation bubble on both sides. Within the central separation two foci are observed, located symmetrically about the centreline and qualitatively in agreement with the oil-streak images of [11]. Figure 9 (b) shows the same picture for one of the sidewalls at $x_1 = 0.14$, in which the large sidewall reversed flow can be seen. The streamlines initially follow the deflected flow of the incident swept shock, and flatten out above $x_1 \sim 50$ to follow the freestream towards the outlet. Below $x_1 \sim 50$ the reflected shock interaction causes a severe reversal of the velocity streamlines, with reverse sidewall flow seen for almost a third of the domain height.



(a) Instantaneous skin-friction distribution on the sidewall. The dashed white line highlights the zero crossing.



(b) Instantaneous skin-friction distribution on the bottom wall. The dashed white line highlights the zero crossing.

Figure 8: Side (a) and bottom wall (b) skin-friction for the 3D laminar SBLI with sidewall effects.

6 3D Shockwave/boundary-layer interaction with forced transition

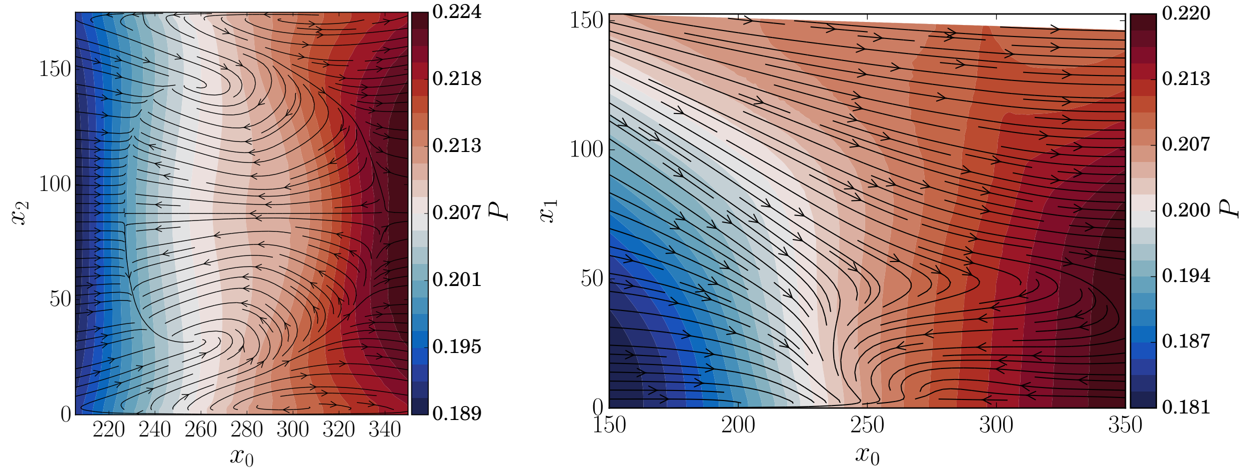
In this section disturbances are added to the laminar base flow, to investigate the transition mechanism for 3D SBLI with sidewall effects. Disturbances are generated by a strip of time-dependent blowing/suction upstream of the interaction region, forced at an unstable mode obtained from linear stability analysis of a Mach 2 laminar separation bubble [34].

6.1 Forcing mechanism

Modal time-dependent forcing is applied as a piecewise condition on the no-slip wall such that the wall-normal velocity is modified in a strip between $10 < x_0 < 30$ as

$$u_1 = \begin{cases} A_0 \cos\left(\beta x_2\right) \sin\left(\omega t\right) \exp\left(-\left(\frac{(x_0-20)^2}{B}\right)\right) & \text{if } 10 < x_0 < 30 \\ 0 & \text{otherwise.} \end{cases}$$

The constants $A_0 = 5 \times 10^{-2}$, $\beta = \frac{5\times 2\pi}{Lx_2} = 0.195$, $\omega = 0.056$ and $B = 4$ are the forcing amplitude, spanwise wavenumber, forcing frequency and variance of the Gaussian respectively. Numerical values of the constants are taken as the most unstable mode obtained by linear stability theory of a Mach 2 laminar separation bubble from [34]. The spanwise wavenumber β is selected to fit five whole wavelengths across the span when



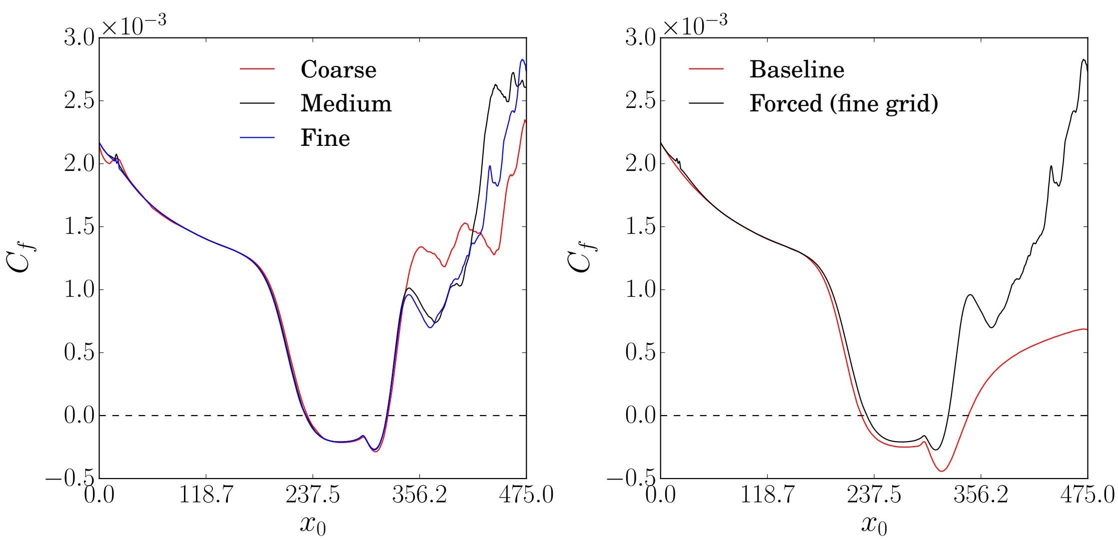
(a) Bottom wall velocity streamlines, $x_1 = 0.14$.

(b) Sidewall velocity streamlines, $x_2 = 0.14$.

Figure 9: Pressure contours and velocity streamlines at $x_1, x_2 = 0.14$ above the bottom wall (x_0, x_2) (a) and sidewall (x_0, x_1) (b), within the main regions of reverse flow on each.

applied to the bottom wall of the domain, and three wavelengths in the case of the forced sidewalls.

6.2 Bottom wall forced transition



(a) Grid sensitivity of centreline skin-friction.

(b) Effect of forcing on the separation length.

Figure 10: Streamwise skin-friction distribution along the bottom wall, evaluated at the centre of the span. The forced cases are time-averaged over one period of the forcing.

The forcing mechanism was first applied to the bottom wall of the domain; the laminar base flow was restarted and advanced in time for an additional five flow-through times up to a final simulation time of $t = 12.5 \times 10^4$. To assess the sensitivity of the forced solution to grid resolution, forcing was applied to all three grids defined in Table 2. To obtain mean profiles the forced simulation result was averaged over one additional period of the forcing. Figure 10 (a) shows the mean skin-friction distribution on the bottom wall along the centreline. A large rise in skin-friction is observed for all three grid resolutions, where the flow has transitioned downstream of the separation bubble. The medium and fine grids give a similar prediction for mean skin-friction during the onset of the transition, but later diverge as the symmetry of the problem is broken further downstream. The coarse grid is under-resolved, showing the same issues near the inlet as in Figure 6 (a), and disagreement with the two finer grids at the start of transition. Figure 10 (b) demonstrates the effect that the transition has on the centreline skin-friction compared to the laminar case of the previous section. The separation bubble has shortened, most significantly on the reattachment side of the separation bubble. The small increase in skin-friction near the inlet is due to the strip of wall forcing that generates the disturbances.

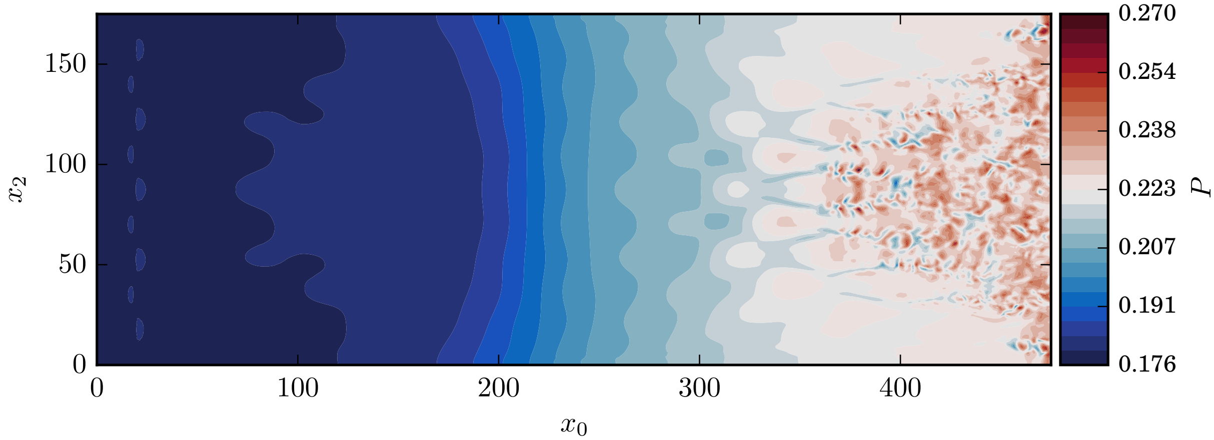


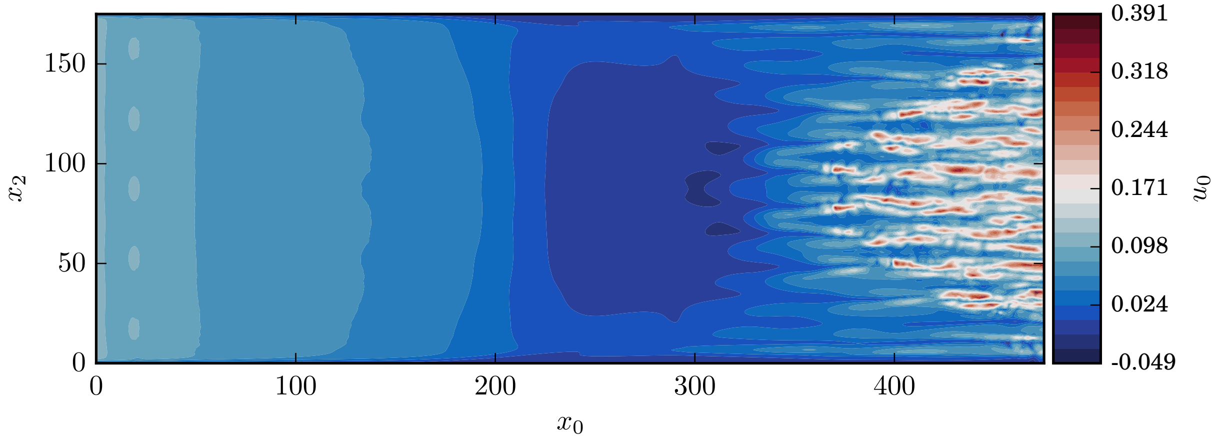
Figure 11: Instantaneous pressure along the bottom wall, with bottom wall forcing induced transition.

The transition is better visualised in Figure 11, which displays instantaneous pressure along the bottom wall. Five wavelengths of forcing can be seen near the inlet, which leads to distortion of the pressure contours upstream of the interaction region. The flow first becomes unstable directly behind the central separation bubble, and is symmetric up until around $x_0 = 400$. In the corner of the domain the separation does not lead to the same magnitude of unsteady behaviour, and does not deviate substantially from the laminar solution until the main transition has spread out across the span at the outlet. The instantaneous streamwise velocity plot of Figure 12 (a) shows a similar picture, unsteady high speed streaks are seen directly downstream of the central separation. The transition is strongest and occurs first at the centreline, and is reduced in streamwise extent behind the regions of attached flow either side of the separation bubble. The high velocity streaks are clearer to see in the time-averaged streamwise velocity contours of Figure 12, showing an alternating pattern of low and high-speed flow. Weaker streaks can be seen in the corner region, that are far less developed than those behind the main interaction. In total there are ten distinct streaks being formed for the five wavelengths of forcing introduced to the system, suggesting that each wave and trough of the wall-normal forcing is generating one of these structures.

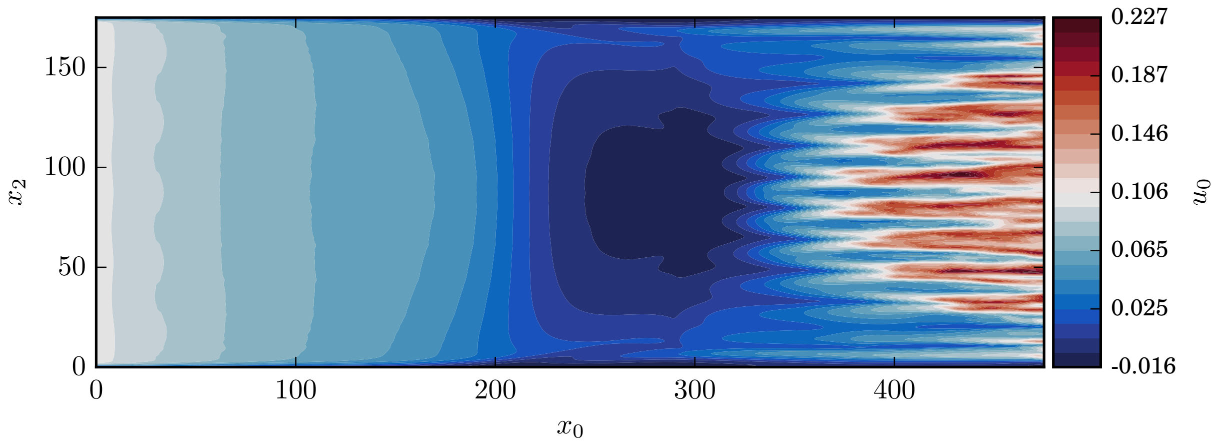
Figure 13 shows the wall-normal vorticity component ω_{x_1} in the transition region for the levels $\omega_{x_1}^{\pm} = 0.05$, coloured by streamwise velocity u_0. A series of hairpin vortices develop independently at the reattachment location of the separation bubble, before rolling up and forming the high velocity streaks. These structures are consistent with the twin prongs seen at the start of each of the high speed streaks in Figure 12 (b). The region of attached flow in Figure 13 between the sidewall and the central separation is seen to transition much later downstream, suggesting that the separation bubble is the main factor in the amplification of the upstream disturbances.

6.3 Sidewall forced transition

The final simulation takes the forcing strip method of the previous section and instead applies it onto both sidewalls, to investigate the susceptibility of the sidewall flow to transition to turbulence. To avoid directly disturbing the generation of the incident shockwave, the forcing was restricted to only 3 wavelengths up the wall from the corner point on either side. This results in a forcing strip applied between $20 < x_0 < 30$ and



(a) Instantaneous streamwise u_0 velocity evaluated at $x_1 = 0.14$.



(b) Time-averaged streamwise u_0 velocity evaluated at $x_1 = 0.14$.

Figure 12: Bottom wall instantaneous (a) and time-averaged (b) velocity u_0 evaluated at $x_1 = 0.14$.

$0 < x_1 < 105$, with the same frequency and amplitude as before. The laminar base flow was again taken from $t = 1.0 \times 10^4$ to $t = 1.25 \times 10^4$, corresponding to five additional flow-through times of the domain.

Figure 14 (a) shows instantaneous pressure contours at the sidewall, with a large region of transitioned flow observed downstream of the sidewall separation. As in the forced bottom wall case, the transition occurs directly behind the central separation and not in the thin corner separation. The disturbances spread out as they move further downstream, occupying up to half of the sidewall by the time the outlet is reached. Figure 14 (b) shows a similar picture, plotting the instantaneous streamwise velocity u_0 at $x_2 = 0.14$ above the sidewall. The high speed streaks are more irregular than those seen in the forced bottom wall case in Figure 12 (a), with small regions of much higher velocity fluid. Wall-normal vorticity on the sidewall is plotted in Figure 15, coloured by streamwise velocity. The transitional flow is again seen directly downstream of the separation bubble and extends up over half of the sidewall. Unsteady behaviour is localised to the sidewall and does not propagate significantly along the span.

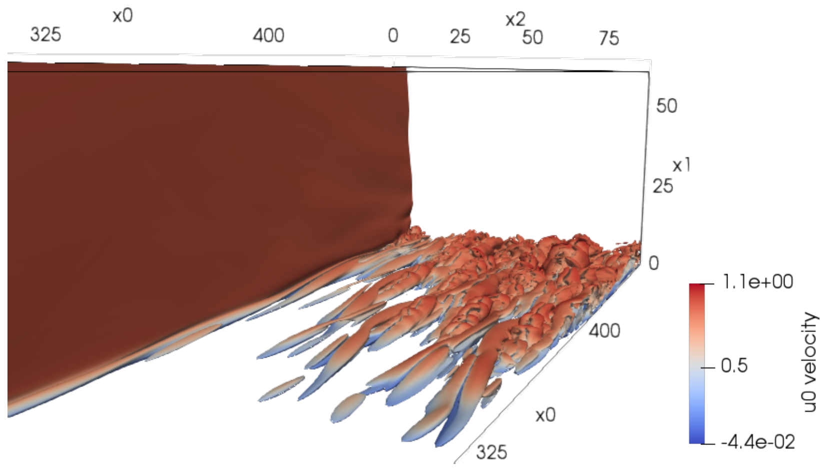


Figure 13: Wall-normal vorticity contours $\omega_{x_1}^{\pm} = \pm 0.05$ coloured by streamwise u_0 velocity; a zoomed in plot of half of the channel for the forced bottom wall 3D SBLI is shown.

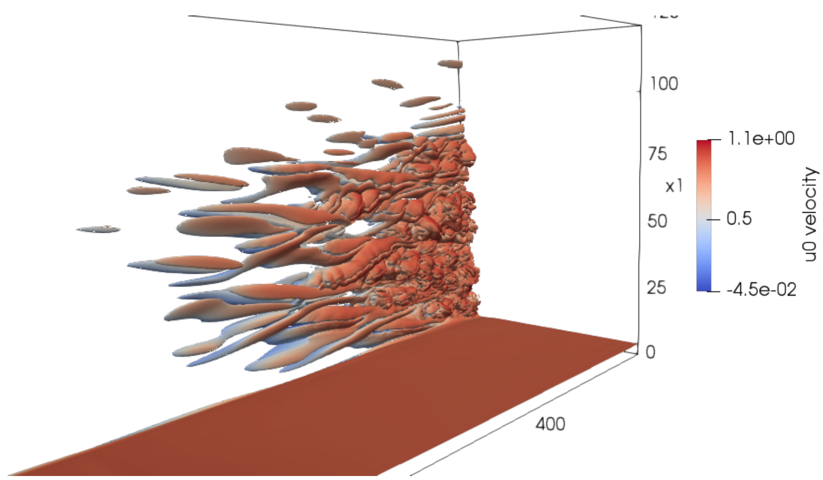
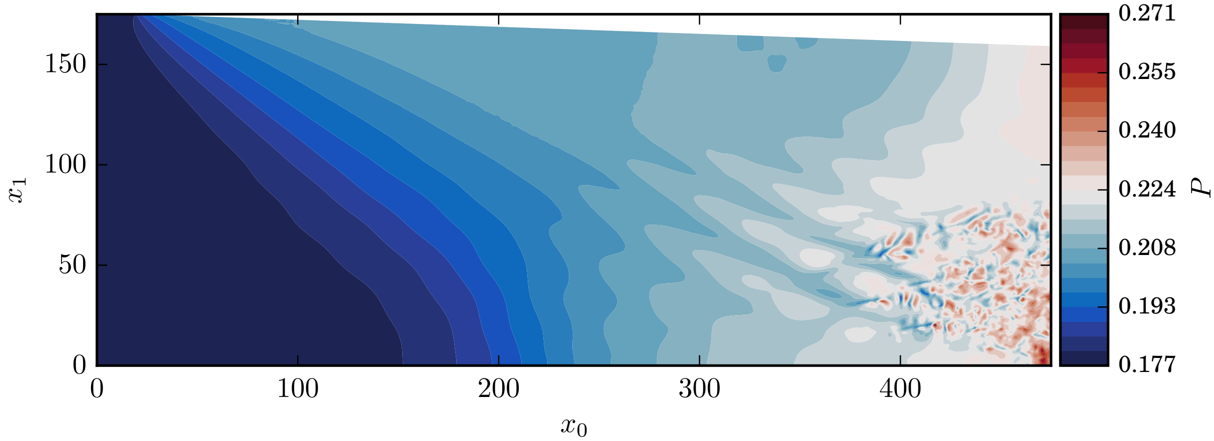


Figure 15: Wall-normal vorticity contours $\omega_{x_2}^{\pm} = \pm 0.05$ coloured by streamwise u_0 velocity; a zoomed in plot of half of the channel for the forced sidewall 3D SBLI is shown.

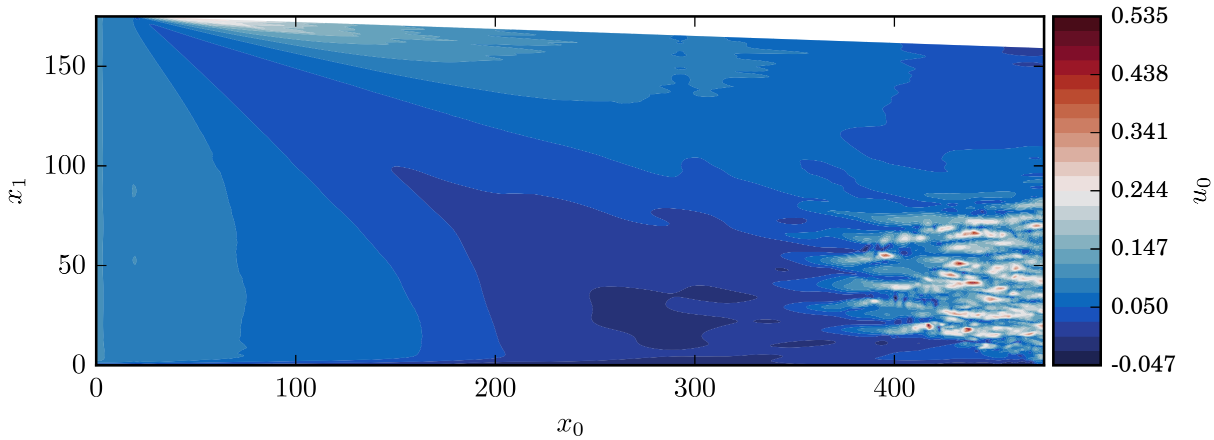
7 Conclusions

Three-dimensional shockwave/boundary-layer interactions (SBLI) at Mach 2 have been simulated for an oblique shock of $\theta = 31.6^\circ$ impinging on a laminar boundary layer. The main contribution of this work has been the inclusion of sidewall confinement effects, to model SBLIs that are often found in internal supersonic duct flows. A stable laminar base flow was obtained, that will form the basis for stability analysis of the system. To demonstrate the susceptibility of the system to a breakdown to turbulence, upstream disturbances were added in the form of a strip of time-dependent forced blowing/suction. A transition to turbulence was observed for disturbances localised to both the bottom and side walls of the domain; in both cases the domain was not long enough for the effect to propagate beyond the regions directly downstream of the forcing.

For the laminar solution a large central separation was seen at the shock impingement point, bordered by thin regions of attached flow. The swept interaction of the incident shock with the sidewall boundary layer caused a thickening of the sidewall boundary layer, and curving of the shock that was most apparent in the near sidewall region in Figure 4. Elongated regions of corner separation were observed, with the onset of corner separation seen far upstream of the central separation (Figure 5). In the presence of sidewalls the



(a) Instantaneous pressure contours near the sidewall, evaluated at $x_2 = 0.14$.



(b) Instantaneous streamwise u_0 velocity near the sidewall, evaluated at $x_2 = 0.14$.

Figure 14: Sidewall pressure (a) and streamwise velocity (b) contours with forcing applied to the sidewall.

incident shock is strengthened near the centreline, causing a stronger centreline separation (Figure 6). In between the corner and central separations the incident shock is weakened and a region of attached flow is present. Large regions of sidewall separation were observed above the corner separation, much larger in size compared to the turbulent solution of [7].

Upstream disturbances were added to both the bottom and sidewalls of the domain, forced at an unstable mode from linear stability theory of a laminar separation bubble. Disturbances added on the bottom wall were amplified by the central separation and became unstable, leading to a transition to turbulence. The weaker separations in the corner region led to transition much further downstream compared to the central separation, while the sidewall flow remained laminar. Similar behaviour was witnessed when disturbances were added to both of the sidewalls, transition was seen downstream of the sidewall separation bubbles and the flow in the centre of the domain remained laminar. Transition to turbulence led to an expected reduction in the length of the central separation bubble (Figure 10 (b)), with the greatest reduction seen around the reattachment side of the bubble. A laminar base flow was obtained for a relatively small shock generator ramp angle of 2°. Further studies should assess the generality of the flow features for higher shock strengths, where transition may occur in the absence of artificial upstream disturbances. Furthermore, the long strip of reverse flow propagating to the outlet in Figure 8 needs to be investigated, flow reversal at an outlet boundary can cause issues for CFD codes. At higher aspect ratios the central separation should tend towards the quasi-2D behaviour, as the influence of the swept sidewall effect reduces.

The sidewall mechanism responsible for the strengthened central separation needs to be investigated. In the experimental studies of [6], [11], the effect was attributed to the crossing location of compression waves emitted at the start of the corner separations. When corner waves from each sidewall crossed at the centreline within the central separation, a strengthening of the separation was observed. Corner waves meeting before the central interaction were said to reduce the size of the interaction, suggesting that flow control applied to the corners of a duct can modify the central separation and subsequent state of the downstream flow as desired. Streamline plots of the reverse flows in Figure 9 (a) and (b) serve as comparison to experimental oil-streak patterns, containing similar structures as seen in the experiments.

8 Acknowledgements

DJL is funded by an EPSRC Centre for Doctoral Training grant (EP/L015382/1). SPJ is supported by a European Commission Horizon 2020 project grant (671571): 'ExaFLOW: Enabling Exascale Fluid Dynamics Simulations'. The authors acknowledge the compute resources used for this work: the 'Cambridge Service for Data Driven Discovery' (CSD3) system operated by the University of Cambridge Research Computing Service (`http://www.hpc.cam.ac.uk`) funded by EPSRC Tier-2 capital grant EP/P020259/1, and the IRIDIS5 High Performance Computing Facility, and associated support services at the University of Southampton. The OpenSBLI code is available at `https://opensbli.github.io`. Data from this report will be available from the University of Southampton institutional repository.

References

[1] Noel T. Clemens and Venkateswaran Narayanaswamy. Low-Frequency Unsteadiness of Shock Wave/-Turbulent Boundary Layer Interactions. *Annual Review of Fluid Mechanics*, 46(1):469–492, 2014.

[2] David S. Dolling. Fifty years of shock-wave/boundary-layer interaction research: What next? *AIAA Journal*, 39(8):1517–1531, aug 2001.

[3] Edgar Katzer. On the lengthscales of laminar shock/boundary-layer interaction. *Journal of Fluid Mechanics*, 206(-1):477, 1989.

[4] Raimo Jaakko Hakkinen, Isaac Greber, Leon Trilling, and SS Abarbanel. The interaction of an oblique shock wave with a laminar boundary layer. 1959.

[5] Andrea Sansica, Neil Sandham, and Zhiwei Hu. Stability and Unsteadiness in a 2D Laminar Shock-Induced Separation Bubble,. 43rd AIAA Fluid Dynamics Conference, 2013.

[6] Xue Xiang and Holger Babinsky. Corner Effects in Oblique Shock Wave/ Boundary Layer Interactions in Rectangular Channels. In *55th AIAA Aerospace Sciences Meeting*, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan 2017.

[7] Bo Wang, Neil. D. Sandham, Zhiwei Hu, and Weidong Liu. Numerical study of oblique shock-wave/boundary-layer interaction considering sidewall effects. *Journal of Fluid Mechanics*, 767:526âĂŞ561, 2015.

[8] Eric Garnier. Stimulated detached eddy simulation of three-dimensional shock/boundary layer interaction. *Shock Waves*, 19(6):479, Sep 2009.

[9] Iván Bermejo-Moreno, Laura Campo, Johan Larsson, Julien Bodart, David Helmer, and John K. Eaton. Confinement effects in shock wave/turbulent boundary layer interactions through wall-modelled large-eddy simulations. *Journal of Fluid Mechanics*, 758:5–62, 2014.

[10] Walter Eagle, James Driscoll, and Jack Benek. Experimental Investigation of Corner Flows in Rectangular Supersonic Inlets with 3D Shock-Boundary Layer Effects. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, jan 2011.

[11] Xue Xiang and Holger Babinsky. An Experimental Study of Corner Flow Control Applied to an Oblique Shock-Wave/Boundary-Layer Interaction. In *2018 AIAA Aerospace Sciences Meeting*, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan 2018.

[12] Shunsuke Koike and Holger Babinsky. Effect of Vortex Generators on Corner Flow Separation Caused by Shock Wave-Boundary-Layer Interaction. In *2018 AIAA Aerospace Sciences Meeting*, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan 2018.

[13] Christian T. Jacobs, Satya P. Jammy, and Neil D. Sandham. OpenSBLI: A framework for the automated derivation and parallel execution of finite difference solvers on a range of computer architectures. *Journal of Computational Science*, 18:12 – 23, 2017.

[14] David J Lusher, Satya P Jammy, and Neil D Sandham. Shock-wave/boundary-layer interactions in the automatic source-code generation framework OpenSBLI. *Computers & Fluids*, In press, 2018.

[15] István Z. Reguly, Gihan R. Mudalige, Michael B. Giles, Dan Curran, and Simon McIntosh-Smith. The OPS Domain Specific Abstraction for Multi-block Structured Grid Computations. WOLFHPC '14, pages 58–67. IEEE Press, 2014.

[16] Uday Bondhugula, Albert Hartono, J. Ramanujam, and P. Sadayappan. A practical automatic polyhedral program optimization system. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, June 2008.

[17] Didem Unat, Xing Cai, and Scott B. Baden. Mint: realizing CUDA performance in 3D stencil methods with annotated C. *Proceedings of the international conference on Supercomputing*, pages 214–224, 2011.

[18] Michael Lange, Navjot Kukreja, Mathias Louboutin, Fabio Luporini, Felippe Vieira, Vincenzo Pandolfo, Paulius Velesko, Paulius Kazakas, and Gerard Gorman. Devito:Towards a Generic Finite Difference DSL Using Symbolic Python. PyHPC '16, pages 67–75. IEEE Press, 2016.

[19] C. Tenaud, E. Garnier, and P. Sagaut. Evaluation of some high-order shock capturing schemes for direct numerical simulation of unsteady two-dimensional free flows. *International Journal for Numerical Methods in Fluids*, 33(2):249–278, 2000.

[20] Christoph Brehm, Michael F. Barad, Jeffrey A. Housman, and Cetin C. Kiris. A comparison of higher-order finite-difference shock capturing schemes. *Computers and Fluids*, 122:184–208, 2015.

[21] Rafael Borges, Monique Carmona, Bruno Costa, and Wai Sun Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227(6):3191–3211, 2008.

[22] Marcos Castro, Bruno Costa, and Wai Sun Don. High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 230(5):1766–1792, 2011.

[23] Lin Fu, Xiangyu Y. Hu, and Nikolaus A. Adams. A family of high-order targeted ENO schemes for compressible-fluid simulations. *Journal of Computational Physics*, 305:333–359, 2016.

[24] Lin Fu, Xiangyu Y. Hu, and Nikolaus A. Adams. Targeted eno schemes with tailored resolution property for hyperbolic conservation laws. *Journal of Computational Physics*, 349:97 – 121, 2017.

[25] Chi-wang Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws Operated by Universities Space Research Association. *ICASE Report*, (97-65):1–78, 1997.

[26] Guang-Shan Jiang and Chi-Wang Shu. Efficient Implementation of Weighted ENO Schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[27] Mark H Carpenter, Jan Nordström, and David Gottlieb. A Stable and Conservative Interface Treatment of Arbitrary Spatial Accuracy. *Journal of Computational Physics*, 365(98):341–365, 1998.

[28] J.H Williamson. Low-storage runge-kutta schemes. *Journal of Computational Physics*, 35(1):48 – 56, 1980.

[29] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society*, 67(221):73–85, 1998.

[30] Z.J. Wang, Krzysztof Fidkowski, RÃľmi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, H.T. Huynh, Norbert Kroll, Georg May, PerâĂŘOlof Persson, Bram Leer, and Miguel Visbal. HighâĂŘorder cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845.

[31] James DeBonis. Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, jan 2013.

[32] Satya P. Jammy, Christian T. Jacobs, and Neil D. Sandham. Performance evaluation of explicit finite difference algorithms with varying amounts of computational and memory intensity. *Journal of Computational Science*, 2016.

[33] Frank White. *Viscous fluid flow*. McGraw-Hill, New York, NY, 2006.

[34] Y. Yao, L. Krishnan, N. D. Sandham, and G. T. Roberts. The effect of mach number on unstable disturbances in shock/boundary-layer interactions. *Physics of Fluids*, 19(5):054104, 2007.