High-Order Methods for Parabolic Equations using a Space-Time Flux Reconstruction Approach


Published in:
AIAA AVIATION 2021 FORUM

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2021 the Author. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
High-Order Methods for Parabolic Equations using a Space-Time Flux Reconstruction Approach

Thomas McCaughtry*, Rob Watson†, Marco Geron‡
Queen’s University Belfast, BT7 1NN, United Kingdom

A novel implementation of the flux reconstruction (FR) approach featuring a hyperbolic reformulation of the governing equations in space-time is presented for viscous linear and non-linear flow problems in both one and two spatial dimensions. The procedure generates high-order accurate schemes — in both space and time — that can analyse diffusion-type equations by recasting second-order equations as first-order systems. Conventional high-order accurate analysis of parabolic equations is severely restricted by limits on time step, which can be avoided by reformulation into a system of hyperbolic equations, with the caveat that only steady solutions may be considered. However, the computation of the resulting system within the space-time FR framework permits the high-order accuracy analysis of unsteady flows with rapid convergence to the steady state, in the pseudo-time sense, within a procedure that can be implemented in a straightforward manner for diffusion-type problems. Eigendecomposition is used to demonstrate that the new systems are hyperbolic in nature for both the 1D and 2D Advection-Diffusion Equations. The development and successful implementation of first-order space-time FR schemes for the 1D and 2D Diffusion Equations is illustrated. It is also verified that the target order-of-accuracy (OOA) is achieved for schemes involving both one and two spatial dimensions. An application of the space-time flux reconstruction approach to the Euler Equations is presented and discussed, with a view to future implementation to the Unsteady Navier-Stokes Equations with similar hyperbolic reformulation of viscous terms.

I. Nomenclature

\[ a = \text{advection velocity in direction of } x \]
\[ b = \text{advection velocity in direction of } y \]
\[ h = \text{characteristic edge length of a mesh} \]
\[ K = \text{total number of solution points within an element} \]
\[ p = \text{gradient of } \phi \text{ in direction of } x \text{ (static pressure in Section } \nabla D) \]
\[ q = \text{gradient of } \phi \text{ in direction of } y \]
\[ t = \text{co-ordinate in the temporal direction} \]
\[ x = \text{co-ordinate in first spatial direction} \]
\[ y = \text{co-ordinate in second spatial direction} \]
\[ \alpha = \text{diffusivity in direction of } x \]
\[ \beta = \text{diffusivity in direction of } y \]
\[ \tau = \text{pseudo-time} \]
\[ \phi = \text{conserved scalar property} \]
\[ \psi = \text{equivalent number of solution points in a 1D element} \]
\[ \varphi = \text{order of a polynomial interpolating through all solution points within each element} \]
\[ ^* = \text{a property within transformed space} \]
\[ n,k = \text{property within an element } \Omega_n, \text{ at a solution point } k \]
\[ i = \text{first derivative with respect to a variable } i \]
\[ ii = \text{second derivative with respect to a variable } i \]

*PhD Student, School of Mechanical and Aerospace Engineering, tmccaughtry01@qub.ac.uk
†Lecturer, School of Mechanical and Aerospace Engineering, r.watson@qub.ac.uk, AIAA Member
‡Lecturer, School of Mechanical and Aerospace Engineering, m.geron@qub.ac.uk, AIAA Member
II. Introduction

Contemporary analysis of turbulent flows is highly dependent on techniques that resolve only the mean flow field — Reynolds-Averaged Navier-Stokes (RANS) based methods — or only the largest turbulent structures, such as Unsteady Reynolds-Averaged Navier-Stokes (URANS) methods. The adoption of higher fidelity methods for analysing such flows (e.g., Large Eddy Simulation (LES)) has been severely limited by the excessive numerical diffusion that can be introduced by the low order-of-accuracy (OOA) schemes which are typically employed in modern CFD analysis. In contrast, high-order methods exhibit much reduced numerical diffusion but have not been widely adopted for several reasons largely related to practical implementation. Finite-difference methods are incompatible with unstructured grids and parallelisation of high-order finite-volume methods across multiple CPUs can be difficult and computationally expensive [1].

The Flux Reconstruction (FR) approach is a discontinuous, finite-element method that generates schemes of arbitrary OOA, and which is both compatible with unstructured grids and readily parallelisable [2]. It may therefore present a possible solution to the problems discussed previously, however some major concerns persist regarding the implementation of temporal discretisation schemes alongside such an approach for some flow problems. The time step of explicit time marching methods typically scales with mesh density, thus in order to maintain numerical stability for fine grids a small time step must be used [3]. The maximum stable time step is further reduced for high-order schemes and thus the computational cost of such simulations is further increased. In contrast, implicit methods do not possess the same stability restriction on time steps, but increasing the number of solution points within each element (associated with increasing OOA) requires the computation of increasingly large Jacobian matrices [4]. This places significant demands on available computer memory, for example the implicit high-order FR method developed by Vanden Hoeck and Lani was reported as having memory usage of 26,100 kB per fifth order element for a 3D analysis involving the Navier-Stokes Equations, compared to 298 kB for an explicit approach [4].

An alternative approach proposed by Yu is to treat space and time as equivalent dimensions within a spatio-temporal domain and use the FR approach to generate schemes with arbitrary OOA in both space and time [5]. There are no restrictions on time step and the solution is advanced using a dual time stepping procedure, utilising a new time-like parameter known as pseudo-time [5].

The technique developed in [5] was not extended to parabolic equations thus the current work seeks to incorporate equations featuring viscous terms into the space-time FR framework. This is achieved by reformulating the original second-order equation, based on the work of Nishikawa, into a system of first-order equations which now possess a hyperbolic character that confers a number of benefits to the space-time FR approach, including [5][6]:

1) Handling of diffusive terms is simplified, both within the conventional and space-time FR approaches, since an additional correction procedure for gradients is no longer required and a single upwind scheme is sufficient
2) Maximum stable steps (in the pseudo-time sense) can be increased by orders of magnitude for unsteady flow problems
3) Solution gradients are computed with the same accuracy as solution variables

The remainder of the paper is organised into the following sections. Section III provides an overview of how parabolic equations may be analysed using conventional FR through correction of solution derivative and flux derivative polynomials. Section IV introduces the key concepts of the 1D+1 space-time FR approach incorporating hyperbolic reformulation for one spatial dimension and verifies functionality through a study into OOA using the Method of Manufactured solutions. Section V extends the method to two spatial dimensions (the 2D+1 approach), repeats the verification process and demonstrates application to the Euler Equations. Conclusions are drawn in Section VI and plans for future work are discussed.

III. Conventional Flux Reconstruction for Parabolic Equations

The original FR approach was developed for advection-type equations only but Huynh later extended this to facilitate diffusion-type equations [7]. Other similar methods for handling diffusive terms have also been used previously, such as the procedure implemented in [8], but the procedure discussed in [7] will be outlined here.

Consider the 1D Diffusion Equation (Eq. 1) containing a second-order derivative in $x$, where flux in the $x$—direction is given as $F = \alpha \phi_x$.

$$\phi_t + (\alpha \phi_x)_x = 0$$  \hspace{1cm} (1)

The domain, $\Omega$, is divided into $N$ distinct elements which can be mapped to a standard element (denoted by $\Omega_s$ and shown in Fig. 1) that uses a separate spatial co-ordinate system [2]. In this co-ordinate system, $\xi$ replaces $x$ and
ranges from $\xi = -1$ to $\xi = +1$ in $\Omega_S$. Within each standard element a polynomial of order $\phi$ interpolates the solution through $K$ solution points (each denoted by $k = 1, 2 \ldots, K$), where for one dimensional space $\phi = K - 1$. In higher dimensions, an appropriate figurate number based on $\psi$ (where $\psi = \phi + 1$, the equivalent number of solution points in a 1D element) is used to determine the number of solution points in the simplex or another element type. For instance, a triangular element where $\psi = 3$, will contain a total of $K = \frac{1}{2}\psi (\psi + 1) = 6$ solution points. Research has indicated that FR schemes can achieve OOA greater than the expected value of $\phi + 1$ (and therefore $\psi$), with such methods described as super accurate [8].

**Fig. 1** Standard 1D element where $K = 3$ (Red circles: solution points, blue triangles: flux points)

A function $\Theta_n$, shown in Eq. [2a] maps a point within the standard element to the corresponding point in the real element, $\Omega_n$, where $x_n$ and $x_{n+1}$ are its outer points [9]. The inverse of $\Theta$ maps a point from the real element to the corresponding point in the standard element (Eq. [2b]).

$$x = \Theta_n(\xi) = \left( \frac{1 - \xi}{2} \right) x_n + \left( \frac{1 + \xi}{2} \right) x_{n+1}$$
$$\xi = \Theta_n^{-1}(x) = 2 \left( \frac{x - x_n}{x_{n+1} - x_n} \right) - 1$$

A discontinuous polynomial, $\phi_n^{\delta D}$, of order $\phi$ interpolates the solution at the $K$ solution points inside each element $\Omega_n$ at a time $t$ (as shown in Fig. 2), where $\delta$ and $D$ indicate that the solution is approximate and discontinuous respectively. Discontinuities are caused by independent interpolation within each element which typically results in different solution values at element interfaces as shown in Fig. 2.

**Fig. 2** Solution Polynomial, $\phi_n^{\delta D}$ (solid red line), in $\Omega_n$ which is discontinuous relative to Solution Polynomials in $\Omega_{n-1}$ and $\Omega_{n+1}$ (dashed red lines)

However, it is convenient to operate within the transformed space of the standard element thus $\phi_n^{\delta D}$ is transformed as follows, where $J_n = (x_{n+1} - x_n)/2$.

$$\hat{\phi}^{\delta D} = \hat{\phi}^{\delta D}(\xi, t) = J_n \phi_n^{\delta D}(\Theta_n(\xi), t)$$

Before the second spatial derivative of the solution can be computed, it is necessary to approximate the first spatial derivative of the solution at the $K$ solution points within each element. Firstly, the approximate, discontinuous solution polynomial, $\phi_n^{\delta D}$ is constructed in each element by interpolating the transformed solution at the $K$ solution points in each element. This is an order $\phi$ polynomial that may take the form of either a Lagrangian Polynomial [2][7][9] or a Vandermonde Matrix [8] for example.

Secondly, $\phi_n^{\delta D}$ is projected to flux points at either end of the standard element, where discontinuities arise between adjacent elements thus the polynomials must be corrected to ensure conservation at flux points. A common interface
solution is computed at either end of each element, denoted the left and right common interface solutions which are referred to as $\hat{\delta}_{n,L}$ and $\hat{\delta}_{n,R}$ respectively. These are computed from the pair of discontinuous solution values at the corresponding flux points using a suitable scheme such as BR2 or Compact Discontinuous Galerkin (CDG). Example interface fluxes are shown in Eq. 4 and Eq. 5 respectively [7]. These are based on a given weight, $\kappa$, that is allocated to each projected solution value with BR2 an average formulation (where $\kappa = 0.5$) while CDG is one-sided (where $\kappa = 1$) [7].

$$\hat{\delta}_{n,L} = \frac{1}{2} \left( \hat{\delta}_{n-1} + \hat{\delta}_{n} \right)$$  \hspace{1cm} (4a)\\
$$\hat{\delta}_{n,R} = \frac{1}{2} \left( \hat{\delta}_{n} + \hat{\delta}_{n+1} \right)$$  \hspace{1cm} (4b)\\
$$\hat{\delta}_{n,L} = \hat{\delta}_{n-1}$$  \hspace{1cm} (5a)\\
$$\hat{\delta}_{n,R} = \hat{\delta}_{n+1}$$  \hspace{1cm} (5b)

The approximate, continuous solution polynomial, $\hat{\delta}_n^C$, can be constructed by adding a correction function to the discontinuous solution such that $\hat{\delta}_n^C$ takes the value of $\hat{\delta}_{n,L}$ and $\hat{\delta}_{n,R}$ at the left and right flux points respectively. However, the continuous polynomial still approximates the original, discontinuous solution at the solution points as shown in Fig. 3.

![Fig. 3 Continuous, Transformed Solution Polynomial $\hat{\delta}_n^C$ (solid red line), in $\Omega_n$, which is continuous relative to Solution Polynomials in $\Omega_{n-1}$ and $\Omega_{n+1}$ (dashed red lines)](image)

Eq. 6 outlines the correction procedure, where $g_L$ and $g_R$ are the left and right correction functions respectively.

$$\hat{\delta}_n^C = \hat{\delta}_n^D + \left( \hat{\delta}_{n,L} - \hat{\delta}_{n,D} \right) g_L + \left( \hat{\delta}_{n,R} - \hat{\delta}_{n,D} \right) g_R$$  \hspace{1cm} (6)

Similarly, the approximate solution derivative polynomial, $(\hat{\delta}_n^D)_\epsilon$, can be determined using the derivatives of the discontinuous solution polynomial and the correction functions as shown in Eq. 7 where the superscript C implies that some correction has taken place. This is because although the solution polynomial is $C^0$-continuous at element boundaries (and therefore conservative in the sense of the solution), the derivative polynomial is not continuous at these locations. This discontinuous polynomial must be corrected using the same procedure as before to ensure that the flux, corresponding to the solution derivative multiplied by diffusivity as shown in Eq. 4 is conserved between elements.

$$\left( \hat{\delta}_n^C \right)_\epsilon = \left( \hat{\delta}_n^D \right)_\epsilon + \left( \hat{\delta}_{n,L} - \hat{\delta}_{n,D} \right) (g_L)_\epsilon + \left( \hat{\delta}_{n,R} - \hat{\delta}_{n,D} \right) (g_R)_\epsilon$$  \hspace{1cm} (7)

Therefore, an additional polynomial $F^D_n$ is defined in Eq. 8 which is the approximate discontinuous flux polynomial in transformed space.

$$\hat{F}^D_n = \alpha \left( \hat{\delta}_n^C \right)_\epsilon$$  \hspace{1cm} (8)

The objective is now to correct $\hat{F}^D_n$ such that it is $C^0$-continuous at flux points, which can be achieved using a similar procedure to before. The first step is to project the discontinuous polynomial to element flux points which
typically results in different values of flux from adjacent elements, similar to Fig. 2 but for interface fluxes rather than interface solutions. Huynh suggests that for computation of the common interface flux, the weight $k$ is reversed in Eq. 5 from what was implemented previously in the CDG scheme, to preserve the isotropic properties of diffusion [7]. The left and right common numerical fluxes are denoted $F_{n,L}^{\delta I}$ and $F_{n,R}^{\delta I}$.

Instead of calculating the approximate, continuous flux polynomial at this stage, it is more convenient to find the flux divergence polynomial since this will ultimately be used in temporal discretisation. Therefore, it is necessary to differentiate the discontinuous flux polynomial $F^{\delta D}$ with respect to $\xi$, through suitable multiplication of $F^{\delta D}$ and the differentiated Lagrangian polynomial or Vandermonde matrix calculated previously. Similarly, the spatial derivative of the correction function, $(g_L)_\xi$ and $(g_R)_\xi$, has already been determined. The approximate flux divergence polynomial can be computed according to Eq. 9 and although itself discontinuous, the correction procedure ensures that flux across element boundaries is $C^0$-continuous.

$$\left( \hat{F}^{\delta C}_n \right)_\xi = \left( \hat{F}^{\delta D}_n \right)_\xi + \left( \hat{F}^{\delta I}_{n,L} - \hat{F}^{\delta D}_{n,L} \right) (g_L)_\xi + \left( \hat{F}^{\delta I}_{n,R} - \hat{F}^{\delta D}_{n,R} \right) (g_R)_\xi$$

(9)

The determination of approximate flux divergence polynomial allows Eq. 10 to be expressed in the semi-discrete form:

$$\left( \hat{F}^{\delta I}_n \right)_\xi = - \left( \hat{F}^{\delta C}_n \right)_\xi$$

(10)

Equation 10 can then be advanced in time using a suitable temporal discretisation approach such as the Forward Euler or a Runge-Kutta Method, with both solution and flux variables conserved at interfaces. However, it should be noted that compared to the relatively straightforward implementation for first-order equations containing only advective terms, this process is somewhat more complicated by the need for two correction procedures [5][6]. Computational cost per element per iteration is increased and thought must be given to the type of interface flux calculation that is most appropriate for the governing equation. For example, methods employed in the computation of numerical fluxes are typically different for advective and diffusive fluxes, which must be combined in a later step.

The presence of viscous terms also has implications on the efficiency of explicit time marching schemes across various approaches in fluid dynamics, not just flux reconstruction. The maximum stable time step of second-order equations normally scales with $O(h^2)$, therefore for even moderately fine meshes the time step must become very small to maintain numerical stability [5][6].

Hyperbolic reformulation was developed by Nishikawa to address this concern through the recasting of a second-order equation into a first-order relaxation system which can accommodate larger time steps that scale with $O(h)$ [5][6]. However, in exchange for the ability to freely set relaxation time ($T_r$) within the system, only steady flow problems can be analysed. Nonetheless, hyperbolic reformulation has been used in conjunction with flux reconstruction previously in [10], but for steady simulations only.

However, this is, highly conveniently, the situation that arises from the space-time approach. The current work seeks to implement both methods together for the development of high-order accurate schemes in space and time for parabolic equations, with maximum stable time steps that scale with $O(h)$ instead of the conventional $O(h^2)$.

### IV. Analysis in One Spatial Dimension

#### A. Reformulation of Governing Equations

Consider the 1D Advection-Diffusion Equation:

$$\phi_t + a \phi_x - \alpha \phi_{xx} = 0$$

(11)

Using the existing space-time FR approach, Eq. 11 is augmented with an additional solution derivative with respect to a new parameter similar to time (henceforth referred to as pseudo-time) as shown in Eq. 12 [3]. Physical time is treated as an additional dimension equivalent to space and the solution can be advanced to the steady state, in the sense of marching the pseudo-time derivative to zero.

$$\phi_t + \phi_t + a \phi_x - \alpha \phi_{xx} = 0$$

(12)

At this stage, the parabolic Eq. 12 is recast as a system of hyperbolic equations as shown in Eq. 13 [5][6]. The new variables introduced are $\rho$, the solution gradient with respect to $x$, and relaxation time of this first-order system, $T_r$. 

5
\begin{align*}
\phi_r &= -\phi_t - a\phi_x + \alpha p_x \\
p_r &= (\phi_x - p) / T_r
\end{align*}

The key concept in the work of Nishikawa is that at the pseudo-steady state (i.e., \( \phi_r = p_r = 0 \)), Eq. [11] is recovered for any arbitrary value of \( T_r \) as shown in Eq. [14][15][16].

\begin{align*}
\phi_t + a\phi_x - \alpha p_x &= 0 \\
p &= \phi_x
\end{align*}

The solving of Eq. [13] instead of Eq. [11] has several useful advantages over the conventional approach to second-order equations discussed in Section III.

1. Rapid convergence to the pseudo-steady state since the maximum stable time step now scales with \( O(h) \) for the first-order system, instead of \( O(h^2) \) previously.
2. Easier computation of common interface fluxes e.g., it is not necessary to have separate schemes for advective and diffusive fluxes and the additional step in the conventional FR approach of correcting the flux derivative is not necessary.
3. Solution gradients, such as viscous stresses or heat fluxes, are computed within the procedure and to the same level of accuracy as the primary solution variable.

Before explaining some practical aspects of the new approach, it is necessary to confirm that Eq. [13] indeed represents a first-order, hyperbolic system. Consider the vector form shown in Eq. [15] where \( \Phi, A, B, \) and \( Q \) are defined in Eq. [16]

\[
\Phi_r + A\Phi_t + B\Phi_x = Q
\]

\[
\Phi = \begin{bmatrix} \phi \\ p \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} a & -\alpha \\ -1/T_r & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 \\ -p/T_r \end{bmatrix}
\]

\( A \) and \( B \) are combined to form a Jacobian matrix, \( A_n \) for the arbitrary face normal vector \((n_t, n_x)\):

\[
A_n = A n_t + B n_x = \begin{bmatrix} n_t + an_x & -an_x \\ -n_x/T_r & 0 \end{bmatrix}
\]

The eigenvalues of this matrix are given in Eq. [18]

\[
\lambda_1, \lambda_2 = \frac{1}{2} \left[(n_t + an_x) \pm \sqrt{(n_t + an_x)^2 + \frac{4\alpha^2 n_x^2}{T_r}}\right]
\]

From Eq. [18] it is clear that for any positive value of \( \alpha \) and \( T_r \) the eigenvalues of this first-order system are both real and distinct, thus it is hyperbolic. It is possible to increase rate of convergence to the pseudo-steady state through selecting an appropriate value of \( T_r \) [5][6]. However, for convenient illustration of space-time fluxes, a value of \( T_r = 1 \) can be used, resulting in the system shown in Eq. [19]

\begin{align*}
\phi_r + \phi_t + a\phi_x - \alpha p_x &= 0 \\
p_r - \phi_x &= -p
\end{align*}

Equation [19] can be expressed in the conservation form shown in Eqs. [20] and [21]. Solution variables are contained within the vector \( \Phi \) and sources are contained within the vector \( S \). Fluxes in the temporal and spatial dimensions are defined within the vectors \( T \) and \( F \) respectively.

\[
\Phi_r + T_t + F_x = S
\]
\[ \Phi = \begin{bmatrix} \phi \\ p \end{bmatrix}, \quad T = \begin{bmatrix} \phi \\ 0 \end{bmatrix}, \quad F = \begin{bmatrix} a\phi - ap \\ -\phi \end{bmatrix}, \quad S = \begin{bmatrix} 0 \\ -p \end{bmatrix} \] (21)

It is straightforward to implement advection-diffusion schemes using this framework where, for the purposes of computation, the character of the system is purely advective, and yet the real diffusive behavior in the \( x \)-direction is maintained as a pair of waves travelling in opposite directions at identical speeds [5].

The correction procedure of the flux reconstruction approach can be conducted simultaneously for both Eq. [19a] and Eq. [19b] without the need for different schemes to compute numerical fluxes at interfaces for advective and diffusive terms. For example, these numerical fluxes are calculated at any given interface flux point using Rusanov’s Flux given in Eq. [22]

\[ F^{\delta I} = \frac{n}{2} \cdot \left( F^{\delta D}_L + F^{\delta D}_R \right) - \frac{s_{max}}{2} \left( \Phi^{\delta D}_R - \Phi^{\delta D}_L \right) \] (22)

where \( n \) is a vector containing the flux point normals and \( s_{max} \) is the maximum absolute value of \( \lambda \) from Eq. [18] given by:

\[ s_{max} = \frac{1}{2} \left[ \left( |n_1| + |an_x| \right) + \sqrt{\left( |n_1| + |an_x| \right)^2 + \frac{4\alpha|n_x|^2}{T_x}} \right] \] (23)

B. Implementation of the 1D+1 Approach

Analysis of a governing equation that is one dimensional with respect to space but two dimensional with respect to the space-time frame of reference is not a novel approach in itself, but it is worthy of further illustration and discussion here in relation to the current work [4]. In many respects, the practicalities of implementing a 1D+1 approach are very similar to that of typical steady 2D problems except the one spatial dimension is replaced by time, with the fluxes updated to reflect Eq. [21] (for the 1D Advection-Diffusion Equation). For a triangular element \( n \) within the space-time domain, these fluxes are shown in Fig. 4.

![Fig. 4 Temporal, spatial, and spatio-temporal fluxes (T, F and \(F^{st}_{n,k}\) respectively) within a 2D space-time triangular element where \( \psi = 3 \)](image)

Since time is now an intrinsic part of the domain, the initial condition is applied as a boundary condition at the \( t = 0 \) boundary, instead of across the entire spatial domain as in conventional analysis (Fig. 5). At spatial boundaries (\( x = 0, 1 \), typical periodic, Dirichlet or Neumann boundary conditions may be imposed. An outflow boundary condition is applied along the opposite face of the domain (\( t = 1 \)), where the final, unsteady solution exists in the sense of physical time. Once the space-time solution is successfully converged to the pseudo-steady state, this can be extracted by projecting the solution to the flux points along this boundary.
C. Verification of the 1D+1 Approach

The 1D Diffusion Equation is considered using the Method of Manufactured Solutions and Eq. [24] on the space-time domain $[0, 1] \times [0, 1]$. In this analysis, $\alpha = 0.1$.

$$\phi^{xx} = (1 - e^{-5t})sin^2(\pi x)$$  \hspace{1cm} (24)

The artificial source term, $S$, for the first equation is:

$$S = \phi_t - \alpha \phi_{xx}$$

$$= 5e^{-5t}sin^2(\pi x) - 2\alpha \pi^2(1 - e^{-5t})cos(2\pi x)$$  \hspace{1cm} (25)

The two equations involved in this analysis solve for the solution, $\phi$, and its spatial gradient, $p$ (equivalent to $\phi_x$). Thus Dirichlet and Neumann boundary conditions must be imposed at $x = 0, 1$, and are computed from Eq. [24]

$$\phi_{t,0} = \phi_{t,1} = 0$$  \hspace{1cm} (26a)

$$p_{t,0} = p_{t,1} = 0$$  \hspace{1cm} (26b)

Schemes from $\psi = 1$ to $\psi = 10$ (where $\psi = \phi + 1$, i.e., the equivalent number of solution points in a 1D element) were implemented on a variety of unstructured, triangular meshes to provide an estimate of achieved OOA [11]. This study was implemented using a modified algorithm developed by Watson et al. [12]. The location of solution points within each element was by no means optimal and could be significantly improved in future work [13].

A measure of the OOA of a scheme is given by the gradient of a least squares fit of a plot of $\log_{10}(L_2 - Error)$ against $\log_{10}(h)$, as shown in Fig. [6][8]. $L_2$-Error was measured using Eq. [27] where $N$ is the total number of elements and $\phi^{xx}$ refers to the exact solution at the given solution point within the space-time domain.

$$L_2 - Error = \sqrt{\frac{\sum_{n=1}^{N} \sum_{k=1}^{K} (\phi_{n,k}^{xx} - \phi_{n,k})^2}{N \times K}}$$  \hspace{1cm} (27)

The measured OOA of each scheme was, with the exception of $\psi = 1$, above the target value of $\psi$ thus there is some evidence to describe these schemes as super accurate [8]. Fig. [6] strikingly highlights the improvement in accuracy that can be realised through the implementation of high-order methods.
The key point to be noted from Fig. 6 is that the respective OOA are applicable across the space-time domain for the Diffusion Equation, thus equally high OOA can be realised in both space and time. The addition of an extra dimension does require further computational effort and it is yet to be determined to what extent this will offset any potential advantages developed through improved time accuracy. This will be the focus of future work.

At this point, it is useful to discuss an individual solution within the space-time domain, which is presented in Fig. 7. The solution is converged to the steady state in the sense of pseudo-time, i.e., $\phi_t = p_t = 0$. However, the unsteady solution in the sense of physical time can be observed over the entirety of the space-time domain as the solution progresses through the time dimension and the value of $t$ increases. The final, unsteady solution, in the sense of physical time, can be extracted from flux points along the $C=1$ boundary as shown in Fig. 8.

On the same mesh, simulation time for high-order methods is substantially longer due to additional computation required for more numerous solution points per element and the need for smaller pseudo-time steps (to maintain numerical stability). However, the solutions generated using high-order schemes will be significantly more accurate as demonstrated in Fig. 6. This trade-off is presented in Fig. 8, which illustrates solutions for selected schemes implemented on a variety of meshes such that the same number of flux points are present at the $t=1$ boundary (the number of mesh points across the space-time domain overall was similar but not identical).

Figure 8 highlights that high-order methods are much more economical since solution accuracy is much improved for the similar numbers of solution and flux points involved in the analysis. It is also evident from Fig. 6 that it would be extremely computationally expensive to use, for example, a 2nd-order scheme to obtain the high levels of accuracy observed in the 5th- and particularly the 10th-order schemes. Further development of this work would seek to determine the cost benefit of space-time FR with hyperbolic reformulation relative to conventional approaches to the analysis of diffusion-type equations, particularly since the introduction of an additional dimension will undoubtedly increase computational expense.

Extension of the approach to other viscous equations is straightforward, as shown for the Viscous Burgers’ Equation in Eq. 28. An example of a simulation utilising this system is shown in Fig. 9.

$$\phi_t + \phi_x + a \phi_x - \alpha p_x = 0 \tag{28a}$$

$$p_t - \phi_x = -p \tag{28b}$$

Figure 9 shows the converged solution to Eq. 28 containing the evolution of the unsteady solution, in the sense of physical time. This is a useful test case for the current work due to the presence of both non-linear and viscous terms, similar to the Navier-Stokes Equations.
V. Analysis in Two Spatial Dimensions

A. Reformulation of Governing Equations

The approach can be extended to two spatial dimensions by introducing another equation corresponding to the solution gradient in the $y$–direction. Consider the 2D Advection-Diffusion Equation given in Eq. [29] for maximum generality, diffusivity is assumed to be anisotropic, taking the value of $\alpha$ in the $x$–direction and $\beta$ in the $y$–direction.
The introduction of pseudo-time derivatives and hyperbolic reformulation once again recasts the original parabolic equation as a system of first-order, hyperbolic equations given in Eq. 30:

\[
\begin{align*}
\phi_t &= -\phi_x - b\phi_y + \alpha p_x + \beta q_y \\
p_t &= (\phi_x - p) / T_r \\
q_t &= (\phi_y - q) / T_r
\end{align*}
\]

As before, it can be shown that this system is hyperbolic by expressing Eq. 30 in a vector form:

\[
\Phi_t + A\Phi_x + B\Phi_y + C\Phi_z = Q
\]

\[
\begin{align*}
\Phi &= \begin{bmatrix} \phi \\ p \\ q \end{bmatrix}, \\
Q &= \begin{bmatrix} 0 \\ -p/T_r \\ -q/T_r \end{bmatrix} \\
A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
B &= \begin{bmatrix} a & -a & 0 \\ -1/T_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
C &= \begin{bmatrix} b & 0 & -\beta \\ 0 & 0 & 0 \\ -1/T_r & 0 & 0 \end{bmatrix}
\end{align*}
\]

A, B and C are combined to form the Jacobian matrix, \( A_n \), for the arbitrary face normal vector \((n_x, n_y, n_z)\):

\[
A_n = An_t + Bn_x + Cn_y = \begin{bmatrix} n_t + an_x + bn_y & bn_x & -n_x/T_r \\ -n_x/T_r & 0 & 0 \\ -n_y/T_r & 0 & 0 \end{bmatrix}
\]

And finally, the non-zero eigenvalues of this matrix are given in Eq. 34:

\[
\lambda_1, \lambda_2 = \frac{1}{2} \left[ (n_t + an_x + bn_y) \pm \sqrt{(n_t + an_x + bn_y)^2 + \frac{4(an^2_x + bn^2_y)}{T_r}} \right]
\]
From Eq. 34, the system will possess real and distinct eigenvalues for any positive values of $\alpha$, $\beta$ and $T_r$, thus it can be described as hyperbolic. Common numerical fluxes are calculated at any given interface flux point using Eq. 22. The maximum absolute value of $\lambda$, $s_{\text{max}}$, from Eq. 34 is given by:

$$s_{\text{max}} = \frac{1}{2} \left( |n_1| + |a_n z| + |b n_j| \right) + \sqrt{\left( |n_1| + |a n x| + |b n y| \right)^2 + \frac{4 \left( \alpha |n_x|^2 + \beta |n_y|^2 \right)}{T_r}}$$  \hspace{1cm} (35)

As before, for convenient definition of space-time fluxes, a value of $T_r = 1$ can be used, resulting in Eq. 36.

$$\phi_r + \phi_t + a \phi_x - \alpha p_x + b \phi_y - \beta q_y = 0$$  \hspace{1cm} (36a)

$$p_r - \phi_t = -p$$  \hspace{1cm} (36b)

$$q_r - \phi_y = -q$$  \hspace{1cm} (36c)

Equation 36 can be expressed in the conservation form shown in Eq. 37 and Eq. 38. Solution variables are contained within the vector $\Phi$ and sources are contained within the vector $S$. Fluxes in the temporal, $x$-, and $y$-directions are defined within the vectors $T$, $F$, and $G$ respectively.

$$\Phi_t + T_t + F_x + G_y = S$$  \hspace{1cm} (37)

$$\Phi = \begin{bmatrix} \phi \\ p \\ q \end{bmatrix}, \quad T = \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix}, \quad F = \begin{bmatrix} a \phi - \alpha p \\ -\phi \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} b \phi - \beta q \\ 0 \\ -\phi \end{bmatrix}, \quad S = \begin{bmatrix} 0 \\ -p \\ -q \end{bmatrix}$$  \hspace{1cm} (38)

As for the one dimensional case, the implementation of the diffusion-type schemes is simplified, requiring only a single upwind and correction procedure. As before, for the purposes of computation the system possesses an advective character but the real diffusive nature is preserved within the pseudo-steady solution. It is important to note that both spatial gradients $p$ and $q$ are computed with the same accuracy as the solution variable $\phi$, which would be useful for the calculation of viscous stresses or heat fluxes when applied to the Navier-Stokes Equations, for instance [5, 6].

B. Implementation of the 2D+1 Approach

The fluxes in Eq. 37 and Eq. 38 are illustrated within a 3D simplex element in Fig. 10. The extension from one spatial dimension to two is therefore straightforward, with viscosity in the second spatial dimension captured within an additional equation.

Similarly to the one dimensional case, the initial condition is applied at the $t = 0$ boundary on the three dimensional space-time domain, and the final, unsteady solution can again be extracted by projecting the solution to flux points on the relevant face, as shown in Fig. 11.

C. Verification of the 2D+1 Approach

Similar to the verification process used for 1D+1 analysis, the 2D Diffusion Equation is considered using the Method of Manufactured Solutions and Eq. 39 on the space-time domain $[0, 1] \times [0, 1] \times [0, 1]$. Isotropic diffusion was assumed, with $\alpha = \beta = 0.1$.

$$\phi^{ex} = (1 - e^{-5t}) \sin^2(\pi x) \sin^2(\pi y)$$  \hspace{1cm} (39)

The artificial source term $S$ for the first element of the source vector $S$ of Eq. 38 is given by Eq. 40.

$$S = \phi_t - \alpha (\phi_{xx} + \phi_{yy})$$

$$= 5e^{-5t} \sin^2(\pi x) \sin^2(\pi y) - 2\alpha \pi^2 (1 - e^{-5t}) \left[ \cos(2\pi x) \sin^2(\pi y) + \sin^2(\pi x) \cos(2\pi y) \right]$$  \hspace{1cm} (40)
Fig. 10 Temporal, spatial (in both the $x$- and $y$-directions) and spatio-temporal fluxes ($T$, $F$, $G$ and $F^{st}$ respectively) within a 3D space-time tetrahedral element where $\psi = 3$ (flux points omitted for clarity)

$T_{n,k} = \begin{pmatrix} \phi_{n,k} \\ 0 \end{pmatrix}$

$F_{n,k} = \begin{pmatrix} a\phi_{n,k} - \alpha p_{n,k} \\ -\phi_{n,k} \end{pmatrix}$

$G_{n,k} = \begin{pmatrix} b\phi_{n,k} - \beta q_{n,k} \\ 0 \\ -\phi_{n,k} \end{pmatrix}$

Fig. 11 3D space-time domain with initial conditions (red) and solution at $t = 1$ (Blue). Solution value, $\phi$, is measured on the t-axis relative to $t = 0$ and $t = 1$ respectively

The three equations involved in this analysis solve for the solution, $\phi$, and its spatial gradients, $p$ and $q$ (equivalent to $\phi_x$ and $\phi_y$ respectively). Thus Dirichlet and Neumann boundary conditions shown in Eq. (41) must be imposed at $x = 0, 1$ and $y = 0, 1$.

\begin{align}
\phi_{t,0,y} &= \phi_{t,1,y} = \phi_{t,x,0} = \phi_{t,x,1} = 0 \\
p_{t,0,y} &= p_{t,1,y} = p_{t,x,0} = p_{t,x,1} = 0 \\
q_{t,0,y} &= q_{t,1,y} = q_{t,x,0} = q_{t,x,1} = 0
\end{align}
Schemes from $k = 1$ to $k = 10$ were implemented on a variety of unstructured, tetrahedral meshes to provide an estimate of achieved OOA, using Eq. [27] similar to the 1D+1 approach [11].

For all 2D+1 schemes, the measured OOA was greater than the target of $\psi$ with a value of approximately $\psi + 1.3$ achieved in general, again suggestive of super accuracy. Fig. [12] further demonstrates the substantial improvements in accuracy that can be realised through the implementation high-order methods.

**Fig. 12** Measured spatio-temporal OOA for 2D+1 Diffusion schemes for $\psi = 1$ to $\psi = 10$

The evolution of the 2D Diffusion Equation through various intervals in physical time is illustrated in Fig. [13]. Convergence in the sense of pseudo-time has been achieved across the space-time domain, yet the unsteady solution which is of equally high-order accuracy in both space and time is available. It should be emphasised that Fig. [13] is a representation of the solution changing with movement along the $t$-axis and thus across the three-dimensional space-time domain. The solution and gradients in $x$ and $y$ are available at every location in both space and time within this mesh.

Again, as for the 1D+1 case, the 2D+1 approach can be readily modified to accommodate other forms of conservation equation, such as the 2D Viscous Burgers’ Equation. Implementation within the space-time framework with reformulation of second-order terms and $T_r = 1$, yields the non-linear, first-order system in Eq. [42]

\[
\begin{align*}
\phi_t + \phi_x + \phi \phi_x - \alpha p_x + \phi \phi_y - \alpha q_y &= 0 \\
p_t - \phi_x &= -p \\
q_t - \phi_y &= -q
\end{align*}
\]

**D. Application to the Euler Equations**

The implementation of the Euler Equations within a space-time flux reconstruction framework was previously demonstrated by Yu [3]. However, it will be briefly discussed here. Consider the 2D Euler Equations shown in Eq. [43] along with Eqs. [44] and [45]

\[
\begin{align*}
\rho_t + \rho u_x + \rho v_y &= 0 \\
\rho u_t + \left(\rho u^2 + p\right)_x + \rho uv_y &= 0 \\
\rho v_t + \rho uv_x + \left(\rho v^2 + p\right)_y &= 0 \\
\rho E_t + (u [\rho E + p])_x + (v [\rho E + p])_y &= 0
\end{align*}
\]
where henceforth, \( \rho, u, v \) and \( E \) refer to fluid density, velocities in the \( x- \) and \( y- \) directions and stagnation energy per unit mass respectively.

\[
c = \sqrt{\frac{\gamma p}{\rho}}
\]

(44)

\[
\rho E = \frac{p}{(\gamma - 1)} + \frac{\gamma}{2} \rho \left( u^2 + v^2 \right)
\]

(45)

where \( c \) is the local speed of sound and \( \gamma \) is the ratio of specific heats.

The addition of a new derivative of the conserved variables with respect to pseudo-time recasts the Euler Equations into a space-time system (Eq. 46), where the derivatives with respect to physical time are now simply temporal fluxes as before.

\[
\begin{align*}
(\rho)_t + (\rho)_x + (\rho u)_x + (\rho v)_y &= 0 \\
(\rho u)_t + (\rho u)_x + \left( \rho u^2 + p \right)_x + (\rho uv)_y &= 0 \\
(\rho v)_t + (\rho v)_x + \left( \rho v^2 + p \right)_y &= 0 \\
(\rho E)_t + (\rho E)_x + (u [\rho E + p])_x + (v [\rho E + p])_y &= 0
\end{align*}
\]

(46a-d)

The Euler Equations possess a hyperbolic character but to confirm this is still the case, the eigendecomposition process is conducted once more. Eq. 46 is expressed in conservation form as shown in Eq. 47.

\[
\Phi_t + \mathbf{T}_t + \mathbf{F}_x + \mathbf{G}_y = 0
\]

(47)

where \( \Phi, \mathbf{T}, \mathbf{F} \) and \( \mathbf{G} \) are conserved solution variables, temporal fluxes and spatial fluxes in the \( x- \) and \( y- \) direction respectively.

These vectors can be defined, in terms of the conserved variables for convenience, as follows:

\[
\Phi = \mathbf{T} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}
\]

(48a)
where eigenvalues shown in eqN USN (48b) is now necessary to apply eqsN TT and TU to revert back from conserved to primitive variables which yields the eigenvalues of matrices L

\[
\begin{bmatrix}
\frac{\phi_2}{\phi_1} (3 - \gamma) + \frac{\phi_2}{\phi_1} (1 - \gamma) + \phi_4 (y - 1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{\phi_3}{\phi_1} (3 - \gamma) + \frac{\phi_3}{\phi_1} (1 - \gamma) + \phi_4 (y - 1)
\end{bmatrix}
\]

(48c)

However, to determine the eigenvalues of this system it is necessary for it to be in vector form, i.e.:

\[
\Phi_x + A \Phi_x + B \Phi_x + C \Phi_y = 0
\]

(49)

where A, B and C are flux jacobian matrices, that are computed as follows:

\[
A = \frac{\partial T}{\partial \Phi}, \quad B = \frac{\partial F}{\partial \Phi}, \quad C = \frac{\partial G}{\partial \Phi}
\]

(50)

Therefore, matrices A, B and C are:

\[
A = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(51a)

\[
B = \begin{bmatrix}
0 & 1 & 0 & 0 \\
\frac{\phi_2}{\phi_1} (y - 3) + \frac{\phi_2}{\phi_1} (y - 1) & \frac{\phi_3}{\phi_1} (3 - \gamma) & \frac{\phi_3}{\phi_1} (1 - \gamma) & \gamma - 1 \\
\frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & 0 \\
\frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & 0
\end{bmatrix}
\]

(51b)

\[
C = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-\frac{\phi_2}{\phi_1} \phi_1 & 0 & \frac{\phi_3}{\phi_1} (3 - \gamma) & \gamma - 1 \\
\frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & 0 \\
\frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & \frac{\phi_1}{\phi_1} \phi_1 & 0
\end{bmatrix}
\]

(51c)

A, B and C are combined to form the Jacobian matrix A, of the system, for the arbitrary face normal vector \((n_x, n_y, n_z)\), as shown previously for the 2D Advection-Diffusion Equation.

The eigenvalues of A, are listed in Eq. 52

\[
\lambda_{1,2} = n_x \phi_2 + n_y \phi_3 + \frac{\sqrt{2}}{2} \phi_1 \phi_2 (n_x^2 + n_y^2) (\gamma - 1) (2 \phi_1 \phi_4 - \phi_2^2 - \phi_3^2)
\]

(52a)

\[
\lambda_{3,4} = n_x \phi_2 + n_y \phi_3 + \frac{\sqrt{2}}{2} \phi_1 \phi_3 (n_x^2 + n_y^2) (\gamma - 1) (2 \phi_1 \phi_4 - \phi_2^2 - \phi_3^2)
\]

(52b)

It is now necessary to apply Eqs. 44 and 45 to revert back from conserved to primitive variables, which yields the eigenvalues shown in Eq. 53.
\[ \lambda_{1,2} = n_t + u n_x + v n_y \]  

\[ \lambda_{3,4} = n_t + u n_x + v n_y \pm c \sqrt{n_x^2 + n_y^2} \]  

The system is hyperbolic since all eigenvalues are real but not strictly hyperbolic since \( \lambda \) is repeated.

A space-time solver similar to that discussed in Section \[V.B\] can be implemented for the Euler Equations, using the fluxes given Eq. \[48\] and a common numerical flux at element interfaces given by the Rusanov’s Flux formulation in Eq. \[22\]. In this case, \( s_{\text{max}} \) is the maximum absolute value of \( \lambda \) from Eq. \[53\] given by Eq. \[54\].

\[ s_{\text{max}} = |n_t| + |u n_x| + |v n_y| + c \sqrt{|n_x|^2 + |n_y|^2} \]  

A simulation using this approach was conducted for a test case involving an isentropic convecting vortex. The space-time domain used in this analysis occupied the region \([0, 10] \times [-5, 5] \times [-5, 5]\) and was divided into tetrahedral elements with periodic boundaries in space \([11]\). The vortex was initially located at \([0, 0, 0]\) with free-stream velocities of \(a = b = 1\).

The converged solution, in the sense of pseudo-time, is shown in Fig. \([14]\) clearly illustrating the path of the vortex through both space and physical time within the space-time domain. It is important to note that while the solution is steady with respect to pseudo-time, the unsteady solution with respect to physical time is presented in Fig. \([14]\) and computed with high order spatial and temporal accuracy.

It is worth emphasizing the superiority of high-order methods in minimizing numerical diffusion. Consider a low-order accurate space-time scheme for the 2D Euler Equations (\(\psi = 2\)) shown in Fig. \([15]\). There is substantial diffusion of the vortex throughout the time axis which is not present in Fig. \([14]\). This is despite there being over fifteen times the number of elements across the space-time domain available to the \(\psi = 2\) scheme (93,750) as for the \(\psi = 6\) scheme (6000) and similar numbers of solution points overall (375,000 and 336,000 respectively). It is clear from Fig. \([6]\) and Fig. \([12]\) that for the \(\psi = 2\) scheme to achieve the same level of accuracy, the degree of computational expense must substantially increase.

On another side note, the number of flux points involved in the low-order accurate analysis (1,125,000) was approximately twice that of the high-order scheme (504,000). This phenomenon associated with high-order methods is highly desirable, since the level of communication required between different CPUs (for parallel operation) when calculating interfaces fluxes is much reduced.

The Navier-Stokes Equations feature viscous terms that are absent from the Euler Equations, thus schemes developed for these are restricted by \(O(h^2)\) time steps. Hyperbolic reformulation can be used, in a similar manner shown in Section \[V.A\] and Section \[V.A\] to develop the original equations into a first-order system as demonstrated previously in \([14],[15]\). Indeed, this approach has been implemented within the flux reconstruction framework by Lou et al. \([10]\). However, such approaches have been largely limited to steady flow problems only thus, future work will seek to accommodate the unsteady Navier-Stokes Equations through the use of space-time flux reconstruction, similar to the current work. The objective would be to achieve high-order accurate spatial and temporal discretisation of the Navier-Stokes Equations with rapid convergence to the pseudo-steady state and other benefits associated with hyperbolic reformulation.

### VI. Conclusion

The space-time FR approach with hyperbolic reformulation provides a novel method for solving diffusion-type problems with high-order accuracy in both space and time. The space-time FR method and hyperbolic reformulation are found to be highly complementary towards each other. For example, first-order systems generated using hyperbolic reformulation of second-order equations, can be analysed with high OOA in both space and time through implementation of a space-time FR approach. Concurrently, hyperbolic reformulation has been shown in previous research to increase the rate of convergence towards the steady state and also simplifies implementation for diffusion-type problems \([5],[6]\).

Verification of the 1D+1 and 2D+1 approaches provide strong evidence that the current method is functioning correctly, where the target OOA of \(\psi\) was achieved for both 1D+1 and 2D+1 schemes. Successful practical implementation for a variety of conservation equations has been demonstrated for very high OOA schemes in both space and time, including the Diffusion Equation and the Euler Equations. However, implementation of the Unsteady Hyperbolic Navier-Stokes Equations would take full advantage of the new approach. This will be the focus of future work but practical application would necessitate extension to three spatial dimensions. Turbulent features are inherently 3D.
effects in reality, thus a 3D+1 space-time formulation of the Navier-Stokes Equations would be highly desirable for the physically correct modelling of such phenomena.

It has been shown that the first-order, space-time systems under consideration have a hyperbolic character. However, it would be beneficial to provide direct evidence that maximum stable pseudo-time steps scale with $O(h)$, associated with hyperbolic equations, as opposed to $O(h^2)$ when second-order derivatives are present. Improvements to the rate of convergence may also be possible through appropriate manipulation of $T_r$, as suggested in \cite{5,6}, instead of the less complex implementation of $T_r = 1$. 

Fig. 14  Space-time solution of the 2D Euler Equations at selected time intervals for $\psi = 6$
Fig. 15  Space-Time Solution of the 2D Euler Equations at selected time intervals for $\psi = 2$

References


