

A Stable and Efficient Hybrid Scheme for Viscous Problems in Complex Geometries

Jing Gong* and Jan Nordström†

January 9, 2007

Abstract

In this paper we present a stable hybrid scheme for viscous problems. The hybrid method combines the unstructured finite volume method with high-order finite difference methods in complex geometries. The coupling procedure between the two numerical methods is based on energy estimates and stable interface conditions are constructed. Numerical calculations show that the hybrid method is efficient and accurate.

keywords Viscous problems, hybrid methods, finite difference, finite volume, coupling procedure, stability, efficiency

1 Introduction

High order finite difference methods (HOFDM) provide an efficient approach when high resolution is essential in a calculation. It is also clear that the unstructured finite volume method (UFVM) is widely used for problems with complex geometries and nonlinear phenomena. In computational physics, the computational domain is often for efficiency and mesh generation reasons divided into multiple blocks, where either HOFDM or UFVM can be used. If a stable and accurate coupling at the block interfaces is achieved we can construct a very flexible and efficient computational method.

In [2], [11], [12] stable interface treatment between multiple domains was presented. However only finite difference methods was used for all blocks. In [14] a hybrid method which combines HOFDM and UFVM for hyperbolic problems on a rather simple geometry was introduced. In [4], various versions of interface procedures for viscous problems in one dimension were investigated. All these methods employ so called summation-by-parts (SBP) operators and impose the boundary conditions weakly, see [1] and [13].

*Dept. of Information Technology, Uppsala University, Sweden

†Dept. of Information Technology, Uppsala University and The Swedish Defence Research Agency, Stockholm, Sweden

In [15], it was shown how to couple the UFVM and HOFDM in a stable way for hyperbolic problems. The energy method and a modification of the dual mesh in the UFVM lead to stability. The present paper continues the study of stable interface treatment by considering hybrid schemes for viscous problems. We also add the additional complexity of a curvilinear mesh in the HOFDM region. The technique derived in this paper makes it straight forward to apply the hybrid technique to the full Navier–Stokes equation.

The rest of the paper is organized as follows. In the next section we derive stable boundary conditions for the continuous problem. Section 3 presents the two numerical methods on a single domain. In Section 4 we derive the stable coupling procedure. In Section 5 numerical experiments are performed. Conclusions are drawn in Section 6.

2 The continuous problem

Consider the model problem

$$u_t + au_x + bu_y = \varepsilon(u_{xx} + u_{yy}), \quad x, y \in \Omega, \quad t > 0, \quad (1a)$$

$$u(x, y, 0) = f(x, y), \quad x, y \in \Omega, \quad (1b)$$

$$\alpha u + \beta \frac{\partial u}{\partial n} = g(x, y, t), \quad x, y \in \partial\Omega, \quad t > 0, \quad (1c)$$

The coefficients a , b and ε are constants. In general, the coefficients α and β depend on x , y , and t .

Let the inner product for real valued functions $u, v \in \Omega$ be defined by $(u, v) = \iint_{\Omega} uv \, dx dy$ and the corresponding norm $\|u\|^2 = (u, u)$. Applying the energy method to (1) yields,

$$\|u\|_t^2 + 2\varepsilon(\|u_x\|^2 + \|u_y\|^2) = - \oint_{\partial\Omega} \left(\bar{c}u^2 - 2\varepsilon u \frac{\partial u}{\partial n} \right) ds. \quad (2)$$

where

$$n = \frac{(dy, -dx)}{ds}, \quad ds = \sqrt{dx^2 + dy^2}, \quad \bar{c} = (a, b) \cdot n, \quad \frac{\partial u}{\partial n} = (u_x, u_y) \cdot n.$$

Substituting the boundary conditions (1c) into (2) we obtain

$$\begin{aligned} \|u\|_t^2 + 2\varepsilon(\|u_x\|^2 + \|u_y\|^2) &= \oint_{\partial\Omega} \left[- \left(\bar{c} + \frac{2\alpha}{\beta} \varepsilon \right) u^2 + \frac{2\varepsilon}{\beta} u g \right] ds \\ &= - \oint_{\partial\Omega} \left(\bar{c} + \frac{2\alpha}{\beta} \varepsilon \right) \left(u - \frac{\varepsilon}{\beta \bar{c} + \frac{2\alpha}{\beta} \varepsilon} g \right)^2 ds \\ &\quad + \oint_{\partial\Omega} \left(\frac{\varepsilon}{\beta} \right)^2 \left(\frac{1}{\bar{c} + \frac{2\alpha}{\beta} \varepsilon} \right) g^2 ds. \end{aligned} \quad (3)$$

This leads to immediately to

Proposition 2.1 *The continuous problem (1) is strongly well posed if*

$$\bar{c} + \frac{2\alpha}{\beta}\varepsilon \geq 0 \text{ on } \partial\Omega. \quad (4)$$

Remark When the solution can be estimated in terms of all types of data, the problem (1) is called strongly well posed, see [5] for more details.

3 The discrete single domain problem

3.1 The finite volume method

The so-called *edge-based* finite volume method is used in this paper (see [3], [6], [8]–[9], [13], and [18] for more details). The computational domain consists of non-overlapping elements and the variables are stored at the nodes of the mesh. For each node, the control volume that constitutes the dual grid is defined as a polygon with its vertices at the centers of gravity of the surrounding triangles (or quadrilaterals) and at the midpoints of the sides, see Figure 1.

In the finite volume method the unknown variable u in equation (1) is discretized by the vector $\mathbf{u} = [u_0, u_1, \dots, u_N]$. \mathbf{u}_x , \mathbf{u}_{xx} , \mathbf{u}_y , and \mathbf{u}_{yy} denote the approximations of u_x , u_{xx} , u_y and u_{yy} , respectively. We define

$$\begin{aligned} \mathbf{u}_x &\approx D_x \mathbf{u} = (P)^{-1} Q_x \mathbf{u}, & \mathbf{u}_{xx} &\approx D_x D_x \mathbf{u} = (P)^{-1} Q_x (P)^{-1} Q_x \mathbf{u}, \\ \mathbf{u}_y &\approx D_y \mathbf{u} = (P)^{-1} Q_y \mathbf{u}, & \mathbf{u}_{yy} &\approx D_y D_y \mathbf{u} = (P)^{-1} Q_y (P)^{-1} Q_y \mathbf{u}, \end{aligned}$$

where P is a positive diagonal matrix with the control volumes Ω_i on the diagonal. In [13], [15] it was shown that the matrices Q_x and Q_y have the components,

$$(Q_x)_{ij} = \frac{dy_j}{2} = -(Q_x)_{ji}, \quad (Q_x)_{ii \notin \partial\Omega} = 0, \quad (Q_x)_{ii \in \partial\Omega} = \frac{dy_i}{2}, \quad (5)$$

$$(Q_y)_{ij} = -\frac{dx_j}{2} = -(Q_y)_{ji}, \quad (Q_y)_{ii \notin \partial\Omega} = 0, \quad (Q_y)_{ii \in \partial\Omega} = -\frac{dx_i}{2}. \quad (6)$$

For the definition of dx_j and dy_j , see Figure 1. Moreover, equations (5) and (6) imply that Q_x and Q_y satisfy

$$Q_x + (Q_x)^T = Y, \quad Q_y + (Q_y)^T = X, \quad (7)$$

where the non-zero elements in Y and X are Δy_i , $-\Delta x_i$ and correspond to the boundary points. Formulas (5)–(6) show that D_x and D_y are Summation-By-Parts (SBP) operators, since

$$(u, v_x) = \int_{\Omega} uv_x \, dx dy = \int_{\Omega} (uv)_x - u_x v \, dx dy = -(u_x, v) + \oint_{\partial\Omega} uv dy,$$

and

$$(\mathbf{u}, \mathbf{v}_x)_P = \mathbf{u}^T P \mathbf{v}_x = \mathbf{u}^T Q_x \mathbf{v} = \mathbf{u}^T (-Q_x^T + Y) \mathbf{v} = -(\mathbf{u}_x, \mathbf{v})_P + \mathbf{u}^T Y \mathbf{v}.$$

See [13] and [15] for more details.

On a grid point i at the boundary $\partial\Omega$, let $(D_n \mathbf{u})_i$ denote an approximation of $\partial u / \partial n$, i.e.,

$$\left(\frac{\partial u}{\partial n}\right)_i = [(u_x, u_y) \cdot n] \approx [(\mathbf{u}_x, \mathbf{u}_y) \cdot \tilde{n}]_i \approx ((D_x \mathbf{u})_i, (D_y \mathbf{u})_i) \cdot \tilde{n}_i = (D_n \mathbf{u})_i, \quad (8)$$

where \tilde{n}_i is the outward pointing normal defined by,

$$\tilde{n}_i = \frac{(dy_i, -dx_i)}{ds_i}, \quad ds_i = \sqrt{dx_i^2 + dy_i^2}, \quad (9)$$

when proceeding counter-clockwise around the domain, see Figure 1(b).

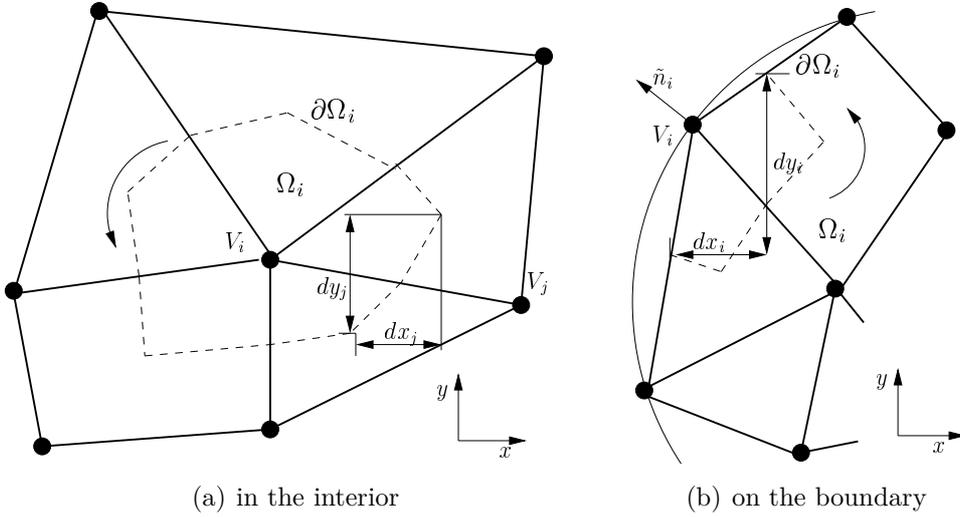


Figure 1: The grid (solid lines) and the dual grid (dashed lines).

A semi-discrete approximation of Equation (1a) can be written,

$$\mathbf{u}_t + aD_x \mathbf{u} + bD_y \mathbf{u} = \varepsilon(D_x D_x \mathbf{u} + D_y D_y \mathbf{u}) + P^{-1}(E_B)^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g] \quad (10)$$

where \mathbf{u}_B represents \mathbf{u} on the boundary $\partial\Omega$. E_B is a projection matrix which maps the values on the computational domain Ω to the outer boundary $\partial\Omega$, that is, $\mathbf{u}_B = E_B \mathbf{u}$ and $(D_n \mathbf{u})_B = E_B (D_n \mathbf{u})$. $\tilde{\alpha}$ and $\tilde{\beta}$ are diagonal matrices where the discrete value of the coefficients α and β are injected on the diagonal, respectively. Γ is a penalty matrix that will be determined below by stability requirements (see [13] and [15]).

By multiplying (10) with $\mathbf{u}^T P$ we obtain,

$$\mathbf{u}^T P \mathbf{u}_t + a \mathbf{u}^T Q_x \mathbf{u} + b \mathbf{u}^T Q_y \mathbf{u} = \varepsilon [\mathbf{u}^T Q_x D_x \mathbf{u} + \mathbf{u}^T Q_y D_y \mathbf{u}] + \mathbf{u}_B^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g]. \quad (11)$$

In equation (11) we note that

$$a\mathbf{u}^T Q_x \mathbf{u} + b\mathbf{u}^T Q_y \mathbf{u} = \frac{1}{2}(a\mathbf{u}^T Y \mathbf{u} + b\mathbf{u}^T X \mathbf{u}) = \frac{1}{2}\mathbf{u}_B^T \Lambda_B \mathbf{u}_B, \quad (12)$$

and

$$\begin{aligned} \mathbf{u}^T Q_x D_x \mathbf{u} + \mathbf{u}^T Q_y D_y \mathbf{u} &= -\mathbf{u}^T (Q_x)^T D_x \mathbf{u} - \mathbf{u}^T Q_y^T D_y \mathbf{u} + \mathbf{u}^T (Y D_x + X D_y) \mathbf{u} \\ &= -(D_x \mathbf{u})^T P(D_x \mathbf{u}) - (D_y \mathbf{u})^T P(D_y \mathbf{u}) + \mathbf{u}_B^T S_B (D_n \mathbf{u})_B. \end{aligned} \quad (13)$$

In (12) and (13) we have introduced (see equations (8) and (9))

$$\bar{c}_i = (a, b) \cdot \tilde{n}_i, \quad \Lambda_B = \text{diag}(\bar{c}_i ds_i), \quad S_B = \text{diag}(ds_i), \quad i \in \partial\Omega$$

We finally obtain,

$$\begin{aligned} \frac{d}{dt} \|\mathbf{u}\|_P^2 + 2\varepsilon \|D_x \mathbf{u}\|_P^2 + 2\varepsilon \|D_y \mathbf{u}\|_P^2 &= -\mathbf{u}_B^T \Lambda_B \mathbf{u}_B + 2\varepsilon \mathbf{u}_B^T S_B D_n \mathbf{u}_B + 2\mathbf{u}_B^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g] \\ &= -\mathbf{u}_B^T (\Lambda_B - 2\Gamma \tilde{\alpha}) \mathbf{u}_B + \underbrace{\mathbf{u}_B^T (2\varepsilon S_B + 2\Gamma \tilde{\beta}) (D_n \mathbf{u})_B}_{(IL)} - 2\mathbf{u}_B^T \Gamma g \end{aligned} \quad (14)$$

In order to obtain an estimate, the term (IL) in (14) must be removed, that is,

$$\Gamma = -\varepsilon S_B \tilde{\beta}^{-1}. \quad (15)$$

Applying the conditions (15) to (14) yields

$$\begin{aligned} \frac{d}{dt} \|\mathbf{u}\|_P^2 + 2\varepsilon \|D_x \mathbf{u}\|_P^2 + 2\varepsilon \|D_y \mathbf{u}\|_P^2 &= -\mathbf{u}_B^T (\Lambda_B + 2\varepsilon S_B \tilde{\beta}^{-1} \tilde{\alpha}) \mathbf{u}_B + 2\varepsilon \mathbf{u}_B^T S_B \tilde{\beta}^{-1} g \\ &= \sum_{i \in \partial\Omega} \left[-\left(\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}} \varepsilon\right) u_i^2 + \frac{2\varepsilon}{\tilde{\beta}_{i,i}} u_i g_i \right] ds_i \\ &= -\sum_{i \in \partial\Omega} \left(\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}} \varepsilon\right) \left(u_i - \frac{\varepsilon}{\tilde{\beta}_{i,i}} \frac{1}{\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}} \varepsilon} g_i\right)^2 ds_i \\ &\quad + \sum_{i \in \partial\Omega} \left(\frac{\varepsilon}{\tilde{\beta}_{i,i}}\right)^2 \left(\frac{1}{\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}} \varepsilon}\right) g_i^2 ds_i. \end{aligned} \quad (16)$$

We have the following proposition.

Proposition 3.1 *If condition (15) and*

$$\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}} \varepsilon \geq 0, \quad i \in \partial\Omega, \quad (17)$$

are satisfied, The problem (10) is strongly stable.

Remark: When the solution can be estimated in terms of all types of data, the problem is called strongly stable, see [5] for more details.

Remark: The estimate (16) is completely similar to the continuous estimate (3).

3.2 The finite difference method

For the finite difference approximations, the physical domain must be possible to smoothly transform to a rectangular computational domain. (see Figure 2). We start by transforming equation (1) to curvilinear form. Note that $u_x = \xi_x u_\xi + \eta_x u_\eta$ and $u_y = \xi_y u_\xi + \eta_y u_\eta$, where we have introduced the transformation $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ and the metric relations,

$$\begin{aligned} J\xi_x &= y_\eta, & J\xi_y &= -x_\eta, & J\eta_x &= -y_\xi, & J\eta_y &= x_\xi, \\ J &= x_\xi y_\eta - x_\eta y_\xi = (\xi_x \eta_y - \xi_y \eta_x)^{-1} \neq 0. \end{aligned}$$

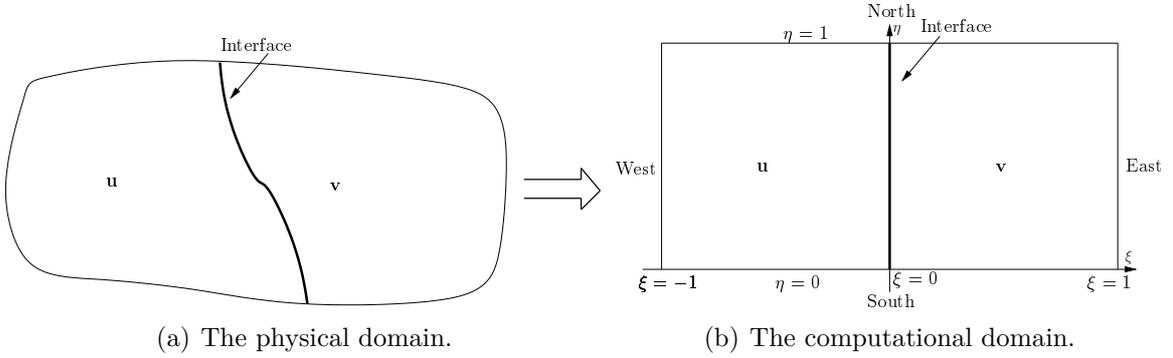


Figure 2:

For simplicity we also introduce the notations,

$$\begin{aligned} \tilde{a} &= aJ\xi_x + bJ\xi_y, & \tilde{b} &= aJ\eta_x + bJ\eta_y, \\ \tilde{f} &= J(\xi_x u_x + \xi_y u_y) = J(\nabla u \cdot \nabla \xi), & \tilde{g} &= J(\eta_x u_x + \eta_y u_y) = J(\nabla u \cdot \nabla \eta). \end{aligned}$$

It follows that

$$J(au_x + bu_y) = (\tilde{a}u)_\xi + (\tilde{b}u)_\eta, \quad J(u_{xx} + u_{yy}) = \tilde{f}_\xi + \tilde{g}_\eta, \quad (18)$$

since $\tilde{a}_\xi + \tilde{b}_\eta = 0$. Equation (1) transforms into

$$Ju_t + (\tilde{a}u)_\xi + (\tilde{b}u)_\eta = \varepsilon(\tilde{f}_\xi + \tilde{g}_\eta). \quad (19)$$

For reasons that will become obvious later, we split the terms $(\tilde{a}u)_\xi$ and $(\tilde{b}u)_\eta$ in (19) as (see [10])

$$(\tilde{a}u)_\xi = \frac{1}{2}(\tilde{a}u)_\xi + \frac{1}{2}\tilde{a}u_\xi + \frac{1}{2}\tilde{a}_\xi u, \quad (\tilde{b}u)_\eta = \frac{1}{2}(\tilde{b}u)_\eta + \frac{1}{2}\tilde{b}u_\eta + \frac{1}{2}\tilde{b}_\eta u.$$

The difference operators in the ξ and η directions on the right subdomain are denoted by $D_\xi = (P_\xi)^{-1}Q_\xi \otimes I_\eta$ and $D_\eta = I_\xi \otimes (P_\eta)^{-1}Q_\eta$, respectively. Note that the operators

$(P_\xi)^{-1}Q_\xi$ and $(P_\eta)^{-1}Q_\eta$ are SBP operators since the matrices P_ξ and P_η are symmetric and positive definite and,

$$\begin{aligned} Q_\xi + (Q_\xi)^T &= B_\xi = \text{diag}([-1, 0, \dots, 0, 1]), \\ Q_\eta + (Q_\eta)^T &= B_\eta = \text{diag}([-1, 0, \dots, 0, 1]). \end{aligned} \quad (20)$$

In matrix formulation we have

$$\begin{aligned} \tilde{\xi}_x &= \text{diag}((\xi_x)_i), & \tilde{\xi}_y &= \text{diag}((\xi_y)_i), \\ \tilde{\eta}_x &= \text{diag}((\eta_x)_i), & \tilde{\eta}_y &= \text{diag}((\eta_y)_i), & \tilde{J} &= \text{diag}(J_i), \\ \tilde{A} &= \text{diag}(\tilde{a}_i), & \tilde{B} &= \text{diag}(\tilde{b}_i), \\ \tilde{F} &= \text{diag}(\tilde{f}_i), & \tilde{G} &= \text{diag}(\tilde{g}_i). \end{aligned}$$

In the curvilinear coordinate system, the finite difference approximation of u at the grid point (ξ_i, η_j) is a vector denoted \mathbf{u}_{ij} . We organize the solution in the global vector $\mathbf{u} = [\mathbf{u}_{11}, \dots, \mathbf{u}_{1l}, \mathbf{u}_{21}, \dots, \mathbf{u}_{2l}, \dots, \mathbf{u}_{n1}, \dots, \mathbf{u}_{nl}]^T$. \mathbf{u}_ξ , \mathbf{u}_η are approximations of u_ξ , u_η and are approximated using the high-order accurate SBP operators for the first derivative that were constructed in [2] and [7]. Moreover, on the boundary we define $D_n \mathbf{u}$ to be the approximation of

$$\begin{aligned} \left(\frac{\partial u}{\partial n} \right)_i &= [(u_x, u_y) \cdot n]_i = [(\xi_x u_\xi + \eta_x u_\eta, \xi_y u_\xi + \eta_y u_\eta) \cdot n]_i \\ &\approx ((\tilde{\xi}_x D_\xi \mathbf{u} + \tilde{\eta}_x D_\eta \mathbf{u})_i, (\tilde{\xi}_y D_\xi \mathbf{u} + \tilde{\eta}_y D_\eta \mathbf{u})_i) \cdot \tilde{n}_i = (D_n \mathbf{u})_i, \quad i \in \partial\Omega, \end{aligned} \quad (21)$$

with

$$\begin{aligned} \tilde{n}_i &= \frac{(dy_i, -dx_i)}{ds_i} = \frac{((y_\xi d\xi + y_\eta d\eta)_i, -(x_\xi d\xi + x_\eta d\eta)_i)}{ds_i}, \\ ds_i &= \sqrt{dx_i^2 + dy_i^2} = \sqrt{(x_\xi d\xi + x_\eta d\eta)_i^2 + (y_\xi d\xi + y_\eta d\eta)_i^2}. \end{aligned} \quad (22)$$

By using the notation above, a semi-discrete approximation of (1) can be written,

$$\begin{aligned} \tilde{J} \mathbf{u}_t + \frac{1}{2} D_\xi (\tilde{A} \mathbf{u}) + \frac{1}{2} \tilde{A} D_\xi \mathbf{u} + \frac{1}{2} (I_\xi \otimes I_\eta) \tilde{A}_\xi \mathbf{u} + \\ \frac{1}{2} D_\eta (\tilde{B} \mathbf{u}) + \frac{1}{2} \tilde{B} D_\eta \mathbf{u} + \frac{1}{2} (I_\xi \otimes I_\eta) \tilde{B}_\eta \mathbf{u} \\ = \varepsilon (D_\xi \tilde{F} + D_\eta \tilde{G}) + \left[(P_\xi)^{-1} \otimes (P_\eta)^{-1} \right] (E_B)^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g] \end{aligned} \quad (23)$$

Here E_B is a projection matrix which maps the values on the computational domain to the outer boundary, that is, $\mathbf{u}_B = E_B \mathbf{u}$ and $(D_n \mathbf{u})_B = E_B (D_n \mathbf{u})$. The boundary conditions have been introduced by using the penalty technique SAT, see [1], [13], and [15].

The energy method leads to

$$\begin{aligned}
& \mathbf{u}^T (P_\xi \otimes P_\eta) \tilde{J} \mathbf{u}_t + \frac{1}{2} \mathbf{u}^T (Q_\xi \otimes P_\eta) \tilde{A} \mathbf{u} + \frac{1}{2} \mathbf{u}^T \tilde{A} (Q_\xi \otimes P_\eta) \mathbf{u} + \frac{1}{2} \mathbf{u}^T (P_\xi \otimes P_\eta) \tilde{A}_\xi \mathbf{u} + \\
& \quad \frac{1}{2} \mathbf{u}^T (P_\xi \otimes Q_\eta) \tilde{B} \mathbf{u} + \frac{1}{2} \mathbf{u}^T \tilde{B} (P_\xi \otimes Q_\eta) \mathbf{u} + \frac{1}{2} \mathbf{u}^T (P_\xi \otimes P_\eta) \tilde{B}_\eta \mathbf{u} \quad (24) \\
& = \varepsilon \mathbf{u}^T (Q_\xi \otimes P_\eta) \tilde{F} + \varepsilon \mathbf{u}^T (P_\xi \otimes Q_\eta) \tilde{G} + \\
& \quad \mathbf{u}_B^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g]
\end{aligned}$$

Remark: Notice that $(P_\xi \otimes P_\eta) \tilde{J}$ is a norm if P_ξ and P_η are diagonal, see Lemma 1 in [12].
Now we can make use of the splitting technique to obtain,

$$\frac{1}{2} \mathbf{u}^T (P_\xi \otimes P_\eta) \tilde{A}_\xi \mathbf{u} + \frac{1}{2} \mathbf{u}^T (P_\xi \otimes P_\eta) \tilde{B}_\eta \mathbf{u} = \frac{1}{2} \mathbf{u}^T (P_\xi \otimes P_\eta) (\tilde{A}_\xi + \tilde{B}_\eta) \mathbf{u} = 0$$

since $\tilde{A}_\xi + \tilde{B}_\eta = \text{diag}((\tilde{a}_\xi + \tilde{b}_\eta)_i) = 0$. We also need,

$$\begin{aligned}
& \frac{1}{2} \mathbf{u}^T (Q_\xi \otimes P_\eta) \tilde{A} \mathbf{u} + \frac{1}{2} \mathbf{u}^T \tilde{A} (Q_\xi \otimes P_\eta) \mathbf{u} = \frac{1}{2} \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{A} \mathbf{u}, \\
& \frac{1}{2} \mathbf{u}^T (P_\xi \otimes Q_\eta) \tilde{B} \mathbf{u} + \frac{1}{2} \mathbf{u}^T \tilde{B} (P_\xi \otimes Q_\eta) \mathbf{u} = \frac{1}{2} \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{B} \mathbf{u}.
\end{aligned}$$

The viscous terms becomes,

$$\begin{aligned}
& \mathbf{u}^T (Q_\xi \otimes P_\eta) \tilde{F} + \mathbf{u}^T (P_\xi \otimes Q_\eta) \tilde{G} \\
& = - \underbrace{\left[(D_\xi \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{F} + (D_\eta \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{G} \right]}_{(\text{Diss})} \\
& \quad + \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{F} + \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{G}
\end{aligned} \quad (25)$$

As was shown above we have,

$$\begin{aligned}
\tilde{F} &= \text{diag}(\tilde{f}) = \text{diag}([J(u_x \xi_x + u_y \xi_y)]_i) = \tilde{J}[(\tilde{\xi}_x^2 + \tilde{\xi}_y^2) D_\xi \mathbf{u} + (\tilde{\eta}_x \tilde{\xi}_x + \tilde{\eta}_y \tilde{\xi}_y) D_\eta \mathbf{u}], \\
\tilde{G} &= \text{diag}(\tilde{g}) = \text{diag}([J(u_x \eta_x + u_y \eta_y)]_i) = \tilde{J}[(\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y) D_\xi \mathbf{u} + (\tilde{\eta}_x^2 + \tilde{\eta}_y^2) D_\eta \mathbf{u}].
\end{aligned}$$

This implies that (Diss) in (25) becomes,

$$\begin{aligned}
(\text{Diss}) &= (D_\xi \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{J}[(\tilde{\xi}_x^2 + \tilde{\xi}_y^2) D_\xi \mathbf{u} + (\tilde{\eta}_x \tilde{\xi}_x + \tilde{\eta}_y \tilde{\xi}_y) D_\eta \mathbf{u}] + \\
& \quad (D_\eta \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{J}[(\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y) D_\xi \mathbf{u} + (\tilde{\eta}_x^2 + \tilde{\eta}_y^2) D_\eta \mathbf{u}] \\
&= \underbrace{\begin{bmatrix} D_\xi \mathbf{u} \\ D_\eta \mathbf{u} \end{bmatrix}^T}_{\mathbf{w}^T} \underbrace{\begin{bmatrix} (P_\xi \otimes P_\eta) \tilde{J}(\tilde{\xi}_x^2 + \tilde{\xi}_y^2) & (P_\xi \otimes P_\eta) \tilde{J}(\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y) \\ (P_\xi \otimes P_\eta) \tilde{J}(\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y) & (P_\xi \otimes P_\eta) \tilde{J}(\tilde{\eta}_x^2 + \tilde{\eta}_y^2) \end{bmatrix}}_H \underbrace{\begin{bmatrix} D_\xi \mathbf{u} \\ D_\eta \mathbf{u} \end{bmatrix}}_{\mathbf{w}} \quad (26)
\end{aligned}$$

The following Lemma is proved in the Appendix.

Lemma 3.2 *The term (Diss) in equations (25) and (27) is positive semi-definite.*

Via the previous analysis, equation (24) is rewritten as

$$\begin{aligned}
& \frac{d}{dt} \|\mathbf{u}\|_{P_\xi \otimes P_\eta}^2 + 2\varepsilon (D_\xi \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{F} + 2\varepsilon (D_\eta \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{G} \\
& = \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{A} \mathbf{u} + \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{B} \mathbf{u} + \\
& \quad 2\varepsilon \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{F} + 2\varepsilon \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{G} + \\
& \quad 2\mathbf{u}_B^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g]
\end{aligned} \tag{27}$$

Note that $\xi = \text{const.}$ at the West and East boundaries and that $\eta = \text{const.}$ at the South and North boundaries (see Figure 2). We have

$$\begin{aligned}
\tilde{n}_i &= \frac{((y_\eta)_i, -(x_\eta)_i) d\eta}{ds_i}, & ds_i &= \sqrt{(x_\eta)_i^2 + (y_\eta)_i^2} d\eta, & i \in \text{West}, \\
\tilde{n}_i &= \frac{((y_\xi)_i, -(x_\xi)_i) d\xi}{ds_i}, & ds_i &= \sqrt{(x_\xi)_i^2 + (y_\xi)_i^2} d\xi, & i \in \text{South}, \\
\tilde{n}_i &= \frac{((y_\eta)_i, -(x_\eta)_i) d\eta}{ds_i}, & ds_i &= \sqrt{(x_\eta)_i^2 + (y_\eta)_i^2} d\eta, & i \in \text{East}, \\
\tilde{n}_i &= \frac{((y_\xi)_i, -(x_\xi)_i) d\xi}{ds_i}, & ds_i &= \sqrt{(x_\xi)_i^2 + (y_\xi)_i^2} d\xi, & i \in \text{North}.
\end{aligned} \tag{28}$$

Consequently, the right-hand-side of (27) can be rewritten as

$$\begin{aligned}
& \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{A} \mathbf{u} + \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{B} \mathbf{u} = \mathbf{u}_B^T \Lambda_B \mathbf{u}_B, \\
& \mathbf{u}^T (B_\xi \otimes P_\eta) \tilde{F} + \mathbf{u}^T (P_\xi \otimes B_\eta) \tilde{G} = \mathbf{u}_B^T S_B (D_n \mathbf{u})_B,
\end{aligned} \tag{29}$$

where

$$\Lambda_B = \text{diag}(\bar{c}_i \bar{d}l_i), \quad S_B = \text{diag}(\bar{d}l_i) \tag{30}$$

with

$$\begin{aligned}
& \bar{c}_i = (a, b) \cdot \tilde{n}_i, \\
& \bar{d}l_i = \begin{cases} -p_i^W ds_i, & d\eta p_i^W = \text{diag}(P_\eta)_i, & i \in \text{West}, \\ -p_i^S ds_i, & d\xi p_i^S = \text{diag}(P_\xi)_i, & i \in \text{South}, \\ p_i^E ds_i, & d\eta p_i^E = \text{diag}(P_\eta)_i, & i \in \text{East}, \\ p_i^N ds_i, & d\xi p_i^N = \text{diag}(P_\xi)_i, & i \in \text{North}. \end{cases}
\end{aligned} \tag{31}$$

The relations (28)–(31) inserted in (27) yields

$$\begin{aligned}
& \frac{d}{dt} \|\mathbf{u}\|_{P_\xi \otimes P_\eta}^2 + 2\varepsilon (D_\xi \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{F} + 2\varepsilon (D_\eta \mathbf{u})^T (P_\xi \otimes P_\eta) \tilde{G} \\
& = \mathbf{u}_B^T \Lambda_B \mathbf{u}_B + 2\varepsilon \mathbf{u}_B^T S_B (D_n \mathbf{u})_B + 2\mathbf{u}_B^T \Gamma [\tilde{\alpha} \mathbf{u}_B + \tilde{\beta} (D_n \mathbf{u})_B - g] \\
& = \mathbf{u}_B^T (\Lambda_B - 2\Gamma \tilde{\alpha}) \mathbf{u}_B + \underbrace{\mathbf{u}_B^T (2\varepsilon S_B + 2\Gamma \tilde{\beta}) (D_n \mathbf{u})_B}_{IR} - 2\mathbf{u}_B^T \Gamma g
\end{aligned} \tag{32}$$

To cancel the term (IR) in equation (32), we require

$$\Gamma = -\varepsilon S_B \tilde{\beta}^{-1}. \quad (33)$$

By employing the same technique as in Section 3.1, we prove the following proposition.

Proposition 3.3 *If condition (33) and*

$$\bar{c}_i + \frac{2\tilde{\alpha}_{i,i}}{\tilde{\beta}_{i,i}}\varepsilon \geq 0, \quad i \in \partial\Omega, \quad (34)$$

are satisfied, The problem (23) is strongly stable.

Remark: By inserting (33) and (34) into (32), we obtain an estimate that is completely similar to (3) and (16).

4 Multiple Domains and Interface Conditions

Without loss of generality, we consider a computational domain which consists of two subdomains. The unknown on the left subdomain is denoted by u and on the right subdomain by v , respectively. The same technique described in the previous section is used here to discretize both u and v . The corresponding notations are also modified by adding superscripts L and R in order to identify the left and right subdomains.

Since the outer boundary treatment has been already discussed, we will only focus on the interface treatment. The coupling of \mathbf{u} and \mathbf{v} as well as the first derivatives $D_1^L \mathbf{u}$ and $D_1^R \mathbf{v}$ at the interface will be treated by using the various forms of the SAT technique.

4.1 The finite volume method

A semi-discrete approximation of (1) on the left part of the computational domain can be written,

$$\begin{aligned} \mathbf{u}_t + aD_x^L \mathbf{u} + bD_y^L \mathbf{u} = & \varepsilon (D_x^L D_x^L \mathbf{u} + D_y^L D_y^L \mathbf{u}) + \\ & (P^L)^{-1} (E_I^L)^T F_1^L (\mathbf{u}_I - \mathbf{v}_I) + \\ & (P^L)^{-1} (E_I^L)^T F_2^L [(D_n^L \mathbf{u})_I + (D_n^R \mathbf{v})_I] + \\ & (P^L)^{-1} (D_n^L)^T (E_I^L)^T F_3^L (\mathbf{u}_I - \mathbf{v}_I) + \text{Pen}_1^L, \end{aligned} \quad (35)$$

where Pen_1^L is the penalty term that imposes the outer boundary conditions weakly. The other three penalty terms on the right hand side will be used to couple the left subdomain calculation to the right subdomain calculation. Note that $(D_n^L \mathbf{u})_I + (D_n^R \mathbf{v})_I$ is small and proportional to the truncation error. \mathbf{u}_I and \mathbf{v}_I are vectors which represent \mathbf{u} and \mathbf{v} (\mathbf{v} is the discrete finite difference solution that will be presented below) on the interface,

respectively. E_I^L is a projection matrix which maps the values on the left computational domain to the interface, that is, $\mathbf{u}_I = E_I^L \mathbf{u}$ and $(D_n^L \mathbf{u})_I = E_I^L (D_n^L \mathbf{u})$. F_1^L , F_2^L , and F_3^L are penalty matrices that will be determined below by stability requirements. $D_n^R \mathbf{v}$ is an approximation of $\partial v / \partial n$ which will be derived in the next section.

By multiplying (35) with $\mathbf{u}^T P^L$ we obtain,

$$\begin{aligned} \frac{d}{dt} \|\mathbf{u}\|_{P^L}^2 + 2\varepsilon (\|D_x^L \mathbf{u}\|_{P^L}^2 + \|D_y^L \mathbf{u}\|_{P^L}^2) = & -\mathbf{u}_I^T \Lambda_I^L \mathbf{u}_I + 2\varepsilon \mathbf{u}_I^T S_I^L D_n^L \mathbf{u}_I \\ & + 2\mathbf{u}_I^T F_1^L (\mathbf{u}_I - \mathbf{v}_I) \\ & + 2\mathbf{u}_I^T F_2^L [(D_n^L \mathbf{u})_I + (D_n^R \mathbf{v})_I] \\ & + 2(D_n^L \mathbf{u})_I^T F_3^L (\mathbf{u}_I - \mathbf{v}_I) \\ & + \text{BT}^L. \end{aligned} \quad (36)$$

where BT^L collects the outer boundary terms (see section 3.1) and

$$\Lambda_I^L = \text{diag}[(a, b) \cdot \tilde{n}_i^L ds_i^L], \quad S_I^L = \text{diag}(ds_i^L), \quad i \in \text{Interface.}$$

4.2 The finite difference method

A semi-discrete approximation of (1) on the right subdomain can be written,

$$\begin{aligned} \tilde{\mathbf{v}}_{\mathbf{v}_t} + \frac{1}{2} D_\xi^R (\tilde{A} \mathbf{v}) + \frac{1}{2} \tilde{A} D_\xi^R \mathbf{v} + \frac{1}{2} (I_\xi \otimes I_\eta) \tilde{A}_\xi \mathbf{v} + \\ \frac{1}{2} D_\eta^R (\tilde{B} \mathbf{v}) + \frac{1}{2} \tilde{B} D_\eta^R \mathbf{v} + \frac{1}{2} (I_\xi \otimes I_\eta) \tilde{B}_\eta \mathbf{v} \\ = \varepsilon D_\xi^R \tilde{F} + \varepsilon D_\eta^R \tilde{G} + \\ \left[(P_\xi^R)^{-1} \otimes (P_\eta^R)^{-1} \right] (E_I^R)^T F_1^R (\mathbf{v}_I - \mathbf{u}_I) + \\ \left[(P_\xi^R)^{-1} \otimes (P_\eta^R)^{-1} \right] (E_I^R)^T F_2^R [(D_n^R \mathbf{v})_I + (D_n^L \mathbf{u})_I] + \\ \left[(P_\xi^R)^{-1} \otimes (P_\eta^R)^{-1} \right] (D_n^R)^T (E_I^R)^T F_3^R (\mathbf{v}_I - \mathbf{u}_I) + \text{Pen}^R. \end{aligned} \quad (37)$$

Here Pen^R is the penalty term for the outer boundary conditions on the right part of the computational domain. E_I^R is a projection matrix which maps the values on the right computational domain to the interface, that is, $\mathbf{v}_I = E_I^R \mathbf{v}$ and $(D_n^R \mathbf{v})_I = E_I^R (D_n^R \mathbf{v})$.

The energy method leads to

$$\begin{aligned} \frac{d}{dt} \|\mathbf{v}\|_{P_\xi^R \otimes P_\eta^R}^2 + 2\varepsilon (D_\xi^R \mathbf{v})^T (P_\xi^R \otimes P_\eta^R) \tilde{F} + 2\varepsilon (D_\eta^R \mathbf{v})^T (P_\xi^R \otimes P_\eta^R) \tilde{G} \\ = \mathbf{v}_I^T P_\eta^R (\tilde{A} \mathbf{v}) - 2\varepsilon \mathbf{v}_I^T P_\eta^R S_I^R D_n^R \mathbf{v}_I + \\ 2\mathbf{v}_I^T F_1^R (\mathbf{v}_I - \mathbf{u}_I) + \\ 2\mathbf{v}_I^T F_2^R [(D_n^R \mathbf{v})_I + (D_n^L \mathbf{u})_I] + \\ 2F_3^R (D_n^R \mathbf{v})_I^T (\mathbf{v}_I - \mathbf{u}_I) + \text{BT}^R, \end{aligned} \quad (38)$$

where BT^R collects the outer boundary terms (for details see section 3.2) and

$$\Lambda_I^R = \text{diag}[(p_i^R d\eta)\tilde{a}_i], \quad S_I^R = \text{diag}(p_i^R ds_i^R), \quad p_i^R d\eta = \text{diag}(P_\eta^R)_i, \quad i \in \text{interface}.$$

4.3 Stable interface treatment

Combining (36) and (38) we have

$$\begin{aligned} & \frac{d}{dt} \|\mathbf{u}\|_{P^L}^2 + \frac{d}{dt} \|\mathbf{v}\|_{P_\xi^R \otimes P_\eta^R J}^2 + 2\varepsilon \|D_x^L \mathbf{u}\|_{P^L}^2 + 2\varepsilon \|D_y^L \mathbf{u}\|_{P^L}^2 \\ & + 2\varepsilon (D_\xi^R \mathbf{v})^T (P_\xi^R \otimes P_\eta^R) \tilde{F} + 2\varepsilon (D_\eta^R \mathbf{v})^T (P_\xi^R \otimes P_\eta^R) \tilde{G} \\ & = -\mathbf{u}_I^T \Lambda_I^R \mathbf{u}_I + 2\varepsilon \mathbf{u}_I^T S_I^L (D_n^L \mathbf{u})_I + 2\mathbf{u}_I^T F_1^L (\mathbf{u}_I - \mathbf{v}_I) \\ & + 2\mathbf{u}_I^T F_2^L [(D_n^L \mathbf{u})_I + (D_n^R \mathbf{v})_I] + 2(D_n^L \mathbf{u})_I^T F_3^L (\mathbf{u}_I - \mathbf{v}_I) + \text{BT}^L \\ & + \mathbf{v}_I^T \Lambda_I^R \mathbf{v}_I - 2\varepsilon \mathbf{v}_I^T S_I^R (D_n^R \mathbf{v})_I + 2\mathbf{v}_I^T F_1^R (\mathbf{v}_I - \mathbf{u}_I) \\ & + 2\mathbf{v}_I^T F_2^R [(D_n^R \mathbf{v})_I + (D_n^L \mathbf{u})_I] + 2(D_n^R \mathbf{v})_I^T F_3^R (\mathbf{v}_I - \mathbf{u}_I) + \text{BT}^R \\ & = \mathbf{x}_I^T M_I \mathbf{x}_I + \text{BT}^L + \text{BT}^R \end{aligned} \quad (39)$$

where

$$\mathbf{x}_I = \begin{bmatrix} \mathbf{u}_I & \mathbf{v}_I & (D_n^L \mathbf{u})_I & (D_n^R \mathbf{v})_I \end{bmatrix}^T$$

and

$$\begin{aligned} M_I' &= \begin{bmatrix} -\Lambda_I^L + 2F_1^L & -F_1^L - F_1^R & 0 & 0 \\ -F_1^L - F_1^R & \Lambda_I^R + 2F_1^R & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ M_I'' &= \begin{bmatrix} 0 & 0 & F_2^L + F_3^L + \varepsilon S_I^L & F_2^L - F_3^R \\ 0 & 0 & -F_3^L + F_2^R & F_2^R + F_3^R - \varepsilon S_I^R \\ F_2^L + F_3^L + \varepsilon S_I^L & -F_3^L + F_2^R & 0 & 0 \\ F_2^L - F_3^R & F_2^R + F_3^R - \varepsilon S_I^R & 0 & 0 \end{bmatrix}, \\ M_I &= M_I' + M_I''. \end{aligned} \quad (40)$$

A sufficient condition for M_I to be a negative semi-definite matrix is that M_I' is a negative semi-definite matrix and $M_I'' = 0$. If the conditions

$$\Lambda_I^L = \Lambda_I^R, \quad F_1^L \leq \Lambda_I^R/2; \quad F_1^R = F_1^L - \Lambda_I^R, \quad (41)$$

are satisfied, M_I' is a negative semi-definite matrix. Moreover, the relations

$$S_I^L = S_I^R, \quad F_2^L + F_3^L + \varepsilon S_I^L = 0, \quad -F_3^L + F_2^R = 0, \quad F_2^R + F_3^R - \varepsilon S_I^R = 0 \quad (42)$$

lead to $M_I'' = 0$.

We have now proved the main proposition of this paper

The interface is discretized using 11 points and each line segment has equal length. If we use a fourth-order accurate SBP operator to approximate the first derivative in equation (46), the new modified dual grid points will be located as in Figure 3. Instead of the original control volumes (see Figure 4(a)), the new control volumes (see Figure 4(b)) should be used in order to guarantee the stability of the hybrid scheme.

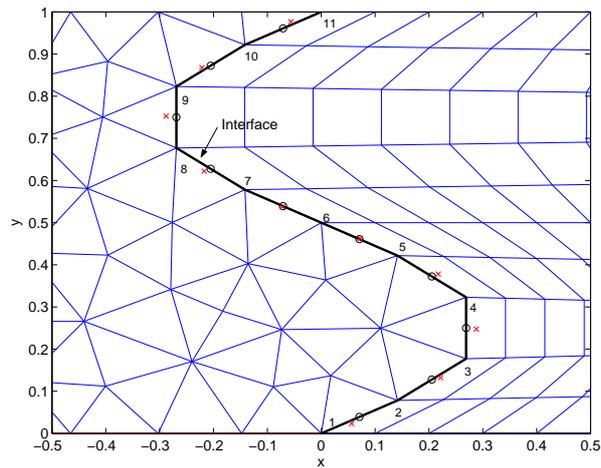


Figure 3: The black 'o' represents the midpoints of the interface and the red 'x' represents the new modified dual grid points.

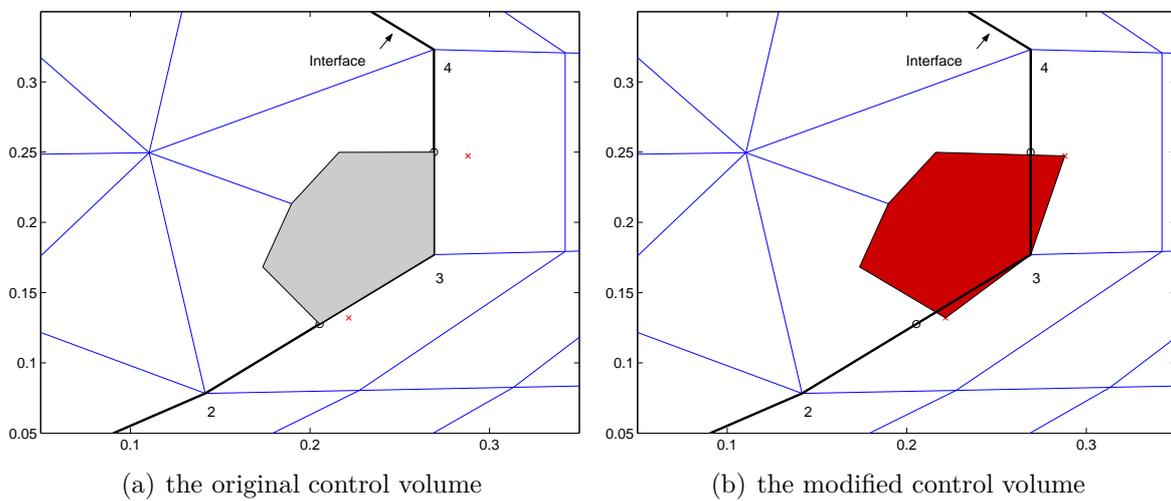


Figure 4: The control volume connected to point 3 at the interface.

5 Numerical Calculations

The model problem tested below is written

$$u_t + au_x + bu_y = \varepsilon(u_{xx} + u_{yy}) + F, \quad (47)$$

with suitable initial data and boundary data. F is the forcing function. In the test we use $a = 1$, $b = 1$, and $\varepsilon = 0.1$. In order to estimate the accuracy of the schemes, an exact solution $u = \sin(2\pi(x + y - 2t))$ has been chosen. The initial data, boundary data and the forcing function F are adjusted to fit the exact solution.

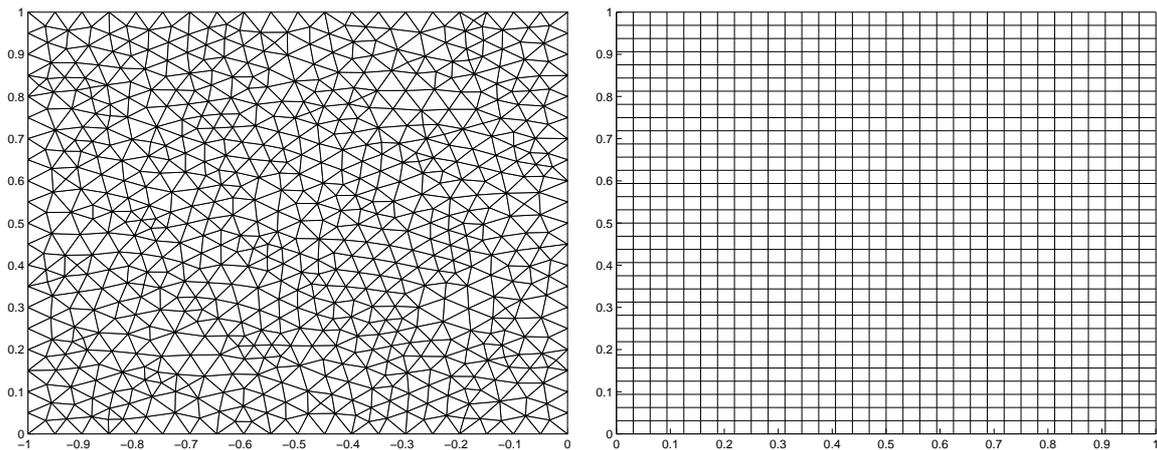
To test the efficiency and accuracy of these schemes, we define the rate of convergence, q , on the computational domain as

$$q = \frac{\log_{10} (\|u - v^{(1)}\|_2 / \|u - v^{(2)}\|_2)}{\log_{10} (\sqrt{N^{(1)}} / \sqrt{N^{(2)}})},$$

where u is the exact solution. $v^{(1)}$ and $v^{(2)}$ are the corresponding numerical solutions on meshes with $N^{(1)}$ and $N^{(2)}$ nodes (including boundary nodes), respectively. We use the classical fourth-order Runge-Kutta method for the time integration. A small time-step is used to minimize the temporal error.

5.1 Single domains and basic accuracy

We start by using the UFVM on unstructured triangle meshes (see Figure 5(a)). The convergence rates are presented in Table 1. Due to symmetry of the triangle meshes, second order accuracy is obtained.



(a) An unstructured mesh with 704 nodes.

(b) A Cartesian mesh with 33×33 nodes.

Figure 5:

Points	\log_{10} -Err	q
182	-1.09	-
704	-1.70	2.06
1607	-2.07	2.07
2807	-2.32	2.06
4357	-2.52	2.09
11139	-2.94	2.07

Table 1: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on a single domain $[-1, 0] \times [0, 1]$ by using the UFVM. Unstructured meshes are used.

Table 2 shows the convergence rates for both UFVM and HOFDM on Cartesian meshes in one computational domain. The nodes of the Cartesian meshes (see Figure 5(b)) are refined from 81 to 66049. The convergence rates for the schemes with interior accuracy of order 2, 4, 6 and boundary accuracy of order 1, 2, 3 are 2, 3, 4th order as shown in [17]. Note that the same error are obtained by using the UFVM and the second order HOFDM. This shows that the UFVM and the second order HOFDM are identical schemes on Cartesian meshes.

Points	UFVM		HOFDM (2nd)		HOFDM (3rd)		HOFDM (4th)	
	\log_{10} -Err	q						
9×9	-0.99	-	-0.99	-	-1.48	-	-	-
17×17	-1.57	2.10	-1.57	2.10	-2.30	2.97	-2.25	-
33×33	-2.17	2.06	-2.17	2.06	-3.16	2.98	-3.34	3.78
65×65	-2.76	2.03	-2.76	2.03	-4.04	3.01	-4.47	3.86
129×129	-3.36	2.02	-3.36	2.02	-4.94	3.01	-5.64	3.93
257×257	-3.97	2.00	-3.97	2.00	-5.83	2.99	-6.83	3.96

Table 2: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on a single domain $[0, 0] \times [1, 1]$ by using the UFVM and HOFDMs. Cartesian meshes are used.

Two more cases for the HOFDM have been tested. Table 3 displays the convergence rate for the HOFDM on stretched meshes (see Figure 6(a)). The stretching coefficients $s_x = (h_x)_{\max}/(h_x)_{\min} = 2$ in x -direction and $s_y = (h_y)_{\max}/(h_y)_{\min} = 3$ in y -direction, where h_x and h_y are the mesh sizes in x -, and y -directions, respectively. From Table 3 we find that correct convergence rates are obtained.

Next, the HOFDM is tested on a sector of an annulus that is given by (see Figure 6(b))

$$x(\xi, \eta) = r \cos(\theta), \quad y(\xi, \eta) = r \sin(\theta), \quad \theta = -\frac{\pi}{4}\xi, \quad r = \frac{1}{2} + \frac{1}{2}\eta.$$

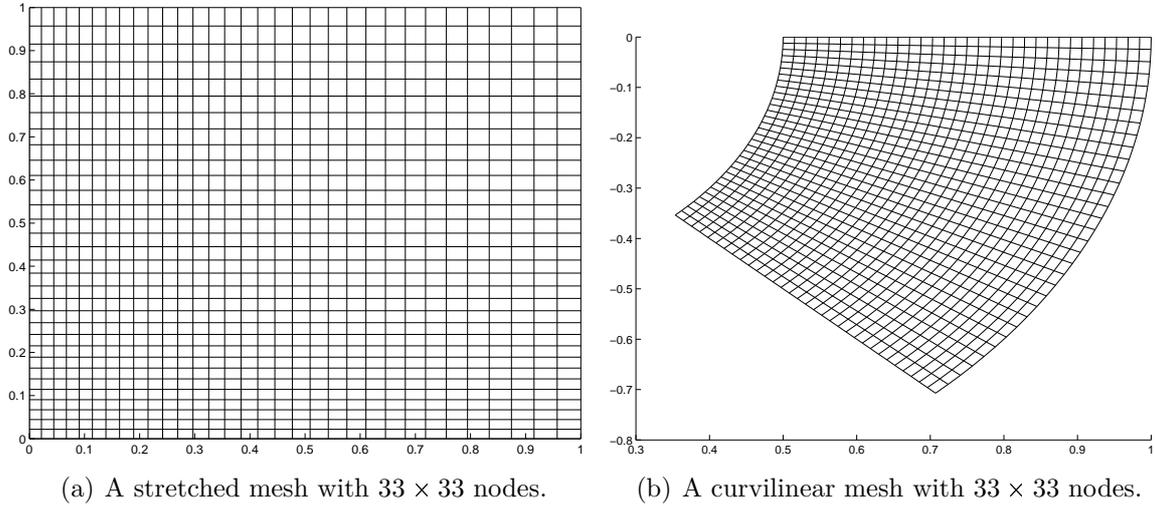


Figure 6:

Points	HOFDM (2nd)		HOFDM (3rd)		HOFDM (4th)	
	\log_{10} -Err	q	\log_{10} -Err	q	\log_{10} -Err	q
9×9	-0.90	—	-1.21	—	—	—
17×17	-1.46	2.03	-1.96	2.70	-1.77	—
33×33	-2.06	2.09	-2.76	2.79	-2.78	3.49
65×65	-2.66	2.05	-3.62	2.92	-3.88	3.75
129×129	-3.27	2.03	-4.51	2.97	-5.03	3.86
257×257	-3.87	2.01	-5.40	2.99	-6.28	3.92

Table 3: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on a single domain $[0, 0] \times [1, 1]$ by using the HOFDM. Stretched meshes are used.

Table 4 shows the convergence rate for all second-, third- and fourth-order HOFDM.

Points	HOFDM (2nd)		HOFDM (3rd)		HOFDM (4th)	
	log ₁₀ -Err	q	log ₁₀ -Err	q	log ₁₀ -Err	q
9 × 9	-1.22	—	-1.78	—	—	—
17 × 17	-1.82	2.16	-2.68	3.25	-2.59	—
33 × 33	-2.41	2.06	-3.63	3.28	-3.77	4.12
65 × 65	-3.01	2.03	-4.58	3.21	-4.89	3.78
129 × 129	-3.61	2.02	-5.51	3.12	-6.02	3.81
257 × 257	-4.21	2.01	-6.43	3.08	-7.17	3.84

Table 4: Convergence rates of approximations to $u_t + u_x + u_y = u_{xx} + u_{yy}$ on a single domain by using the HOFDM. Curvilinear meshes are used.

5.2 Multiple domains

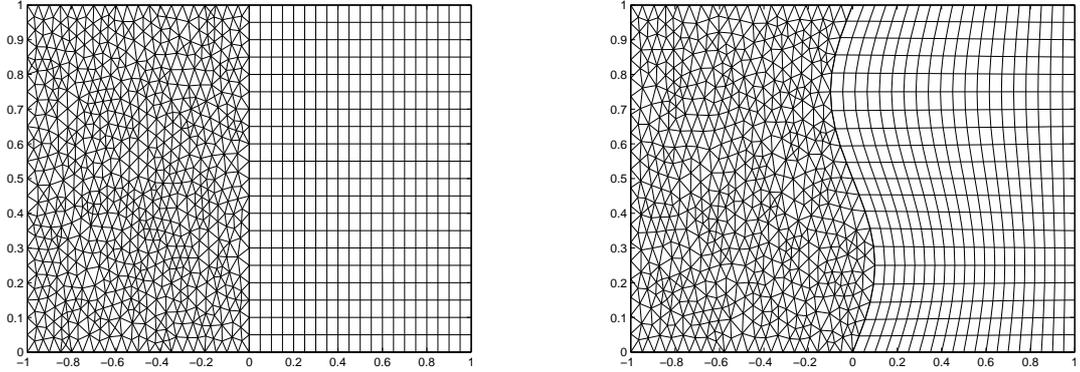
In this section we will illustrate the stability and efficiency of the hybrid scheme on multiple domains. The testing is processed as follows,

1. Applying the UFVM on an unstructured mesh in all subdomains;
2. Using the UFVM on the same mesh in a subdomain and the HOFDM on structured mesh(es) in the other subdomain(s);
3. Adjusting the number of grid points in the subdomain until we obtain a similar L_2 -error in all subdomains.

First we calculate on two subdomains with a linear interface at $x = 0$ (see Figure 7(a)). Table 5 shows the convergence rate of UFVM and second and fourth order accurate HOFDM. The convergence rate for the UFVM is 2 on unstructured symmetrical meshes. The log₁₀ L_2 -error is -3.30 for the UFVM on the finest mesh with 50138 points. We only need a mesh with 28852 points for the hybrid method (UFVM+HOFDM(4th)) to obtain the same error level.

Next, we test the hybrid method on two subdomains with a smooth curved interface (see Figure 7(b)). In the result shown in Table 6, we see that in the fourth order case the hybrid scheme is efficient since only one fifth of the nodes are required for the HOFDM. The error levels are almost same as with a linear interface. The solution and the error are presented in Figure 8. The wave propagates from left to the right via the curved interface without reflection. We conclude that the curved interface does not introduce more error and reflections compared with the linear interface.

Next we will test the hybrid schemes on a computational domain $[-1, 1] \times [-1, 1]$ with four sub-domains (see Figure 9). On the subdomain $[-1, 0] \times [-1, 0]$ excluding an ellipse,



(a) with a linear interface. 702 nodes are used on the left domain and 21×21 nodes are used on the right domain. (b) with a curved interface. 597 nodes are used on the left domain and 21×21 nodes are used on the right domain.

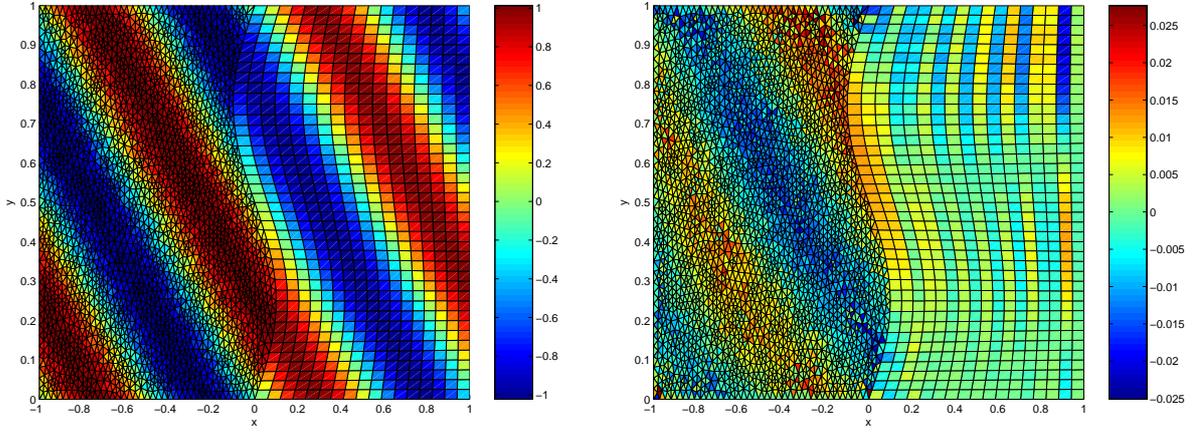
Figure 7: Hybrid mesh with two subdomains

UFVM (Whole domain)			Hybrid(UFVM+HOFDM(2nd))			Hybrid(UFVM+HOFDM(4th))		
Points	\log_{10} -Err	q	Points	\log_{10} -Err	q	Points	\log_{10} -Err	q
360	-1.09	-	292(182 + 110)	-1.09	-	-	-	-
1425	-1.70	2.06	1124(704 + 420)	-1.69	2.05	977(704 + 273)	-1.73	-
3133	-2.07	2.07	2537(1607 + 930)	-2.07	2.16	2072(1607 + 465)	-2.07	2.06
5588	-2.32	2.06	4447(2807 + 1640)	-2.32	2.05	3545(1807 + 738)	-2.32	2.13
8779	-2.52	2.09	6907(4357 + 2550)	-2.52	2.08	5428(4357 + 1071)	-2.53	2.26
22389	-2.94	2.07	17619(11139 + 6480)	-2.94	2.06	13164(11139 + 1863)	-2.93	2.09
50138	-3.30	2.03	39621(25101 + 14520)	-3.30	2.03	28852(25101 + 3751)	-3.30	2.15

Table 5: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on two subdomains with a linear interface. UFVM is used on the left domain and HOFDM is used on the right domain.

UFVM			Hybrid(UFVM+HOFDM(2nd))			Hybrid(UFVM+HOFDM(4th))		
Points	\log_{10} -Err	q	Points	\log_{10} -Err	q	Points	\log_{10} -Err	q
360	-1.09	-	271(160 + 110)	-1.00	-	-	-	-
1425	-1.70	2.06	1017(597 + 420)	-1.62	2.16	870(597 + 273)	-1.64	-
3133	-2.07	2.07	2236(1306 + 930)	-1.98	2.14	1771(1306 + 465)	-1.99	2.26
5588	-2.32	2.06	3942(2302 + 1640)	-2.24	2.10	3040(2302 + 738)	-2.26	2.26
8779	-2.52	2.09	6217(3667 + 2550)	-2.43	1.87	4738(3667 + 1071)	-2.44	1.95
22389	-2.94	2.07	15709(9229 + 6480)	-2.86	2.13	11092(9229 + 1863)	-2.85	2.18
50138	-3.30	2.03	35226(20706 + 14520)	-3.21	2.03	24457(20706 + 3751)	-3.22	2.19

Table 6: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on two subdomains with a curvilinear interface. UFVM is used on the left domain and HOFDM is used on the right domain.



(a) solution at $T = 1.0$.

(b) error at $T = 1.0$.

Figure 8: $\log_{10}(L_2 - error) = -2.22$ on the left subdomain with 2302 nodes and $\log_{10}(L_2 - error) = -2.39$ with 738 nodes for $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$. A curved interface is used.

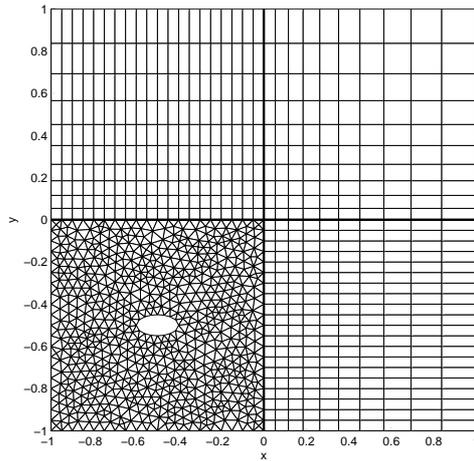


Figure 9: A hybrid mesh with four sub-domains.

the UFVM was used. On the three other subdomains, the HOFDM was used. The finite difference and the finite volume solutions are co-located at the interfaces $y = 0$ and $x = 0$. Table 7 shows the convergence rate by using the hybrid scheme. The solution and error are shown in Figure 10. The efficiency of the hybrid scheme with the 4th order HOFDM is clearly seen.

UFVM (Whole domain)			Hybrid(UFVM+HOFDM(2nd))			Hybrid(UFVM+HOFDM(4th))		
Points	\log_{10} -Err	q	Points	\log_{10} -Err	q	Points	\log_{10} -Err	q
733	-1.09	-	550	-1.11	-	-	-	-
			(187 + 121 + 121 + 121)					
2809	-1.70	2.09	2020	-1.71	2.14	1412	-1.75	-
			(697 + 441 + 441 + 441)			(697 + 273 + 273 + 169)		
6389	-2.07	2.07	4451	-2.08	2.15	2723	-2.09	2.43
			(1568 + 961 + 961 + 961)			(1568 + 465 + 465 + 225)		
11205	-2.32	2.05	7826	-2.34	2.07	4538	-2.35	2.22
			(2783 + 1681 + 1681 + 1681)			(2738 + 738 + 738 + 324)		
17424	-2.52	2.08	12156	-2.54	2.16	6936	-2.56	2.43
			(4353 + 2601 + 2601 + 2601)			(4353 + 1071 + 1071 + 441)		
44447	-2.94	2.06	30721	-2.96	2.07	15285	-2.96	2.21
			(11038 + 6561 + 6561 + 6561)			(11030 + 1863 + 1863 + 529)		
99923	-3.30	2.04	68405	-3.32	2.07	32945	-3.32	2.27
			(24482 + 14641 + 14641 + 14641)			(24482 + 3751 + 3751 + 961)		

Table 7: Convergence rates of approximations to $u_t + u_x + u_y = 0.1(u_{xx} + u_{yy})$ on four sub-domains.

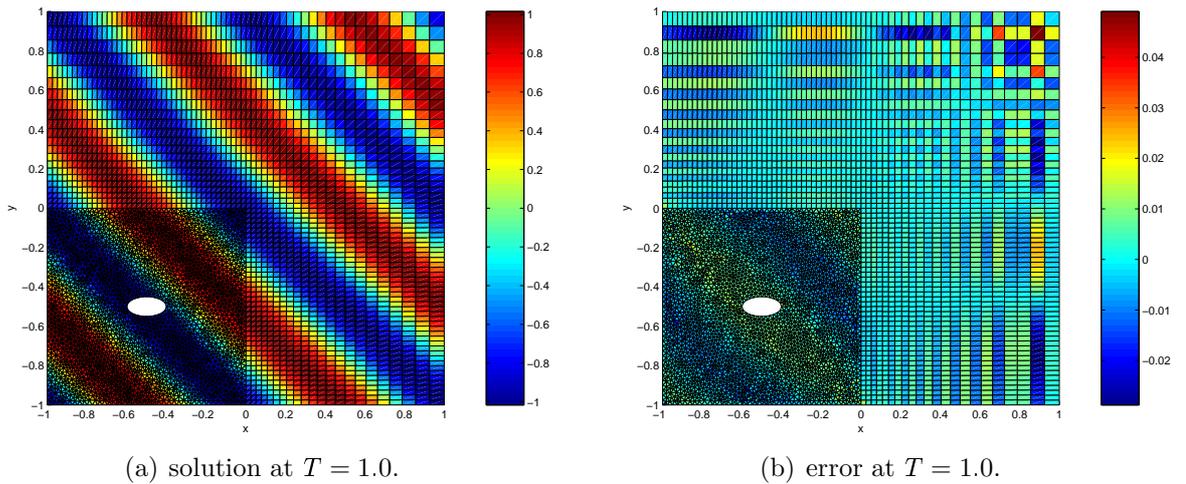


Figure 10: A four sub-domain case.

5.3 Application to heat distribution around rods

Finally we will exemplify our technique by computing the steady heat distribution around a set of rods. Consider the problem,

$$T_t + aT_x + bT_y = \varepsilon(T_{xx} + T_{yy}), \quad -1 \leq x, y \leq 1, \quad t > 0, \quad (48)$$

with a initial condition $T = T_\infty$, and the boundary conditions

$$T = T_b, \quad (a, b) \cdot \hat{n} < 0, \quad \frac{\partial T}{\partial \hat{n}} = 0, \quad (a, b) \cdot \hat{n} \geq 0,$$

at the far-field boundary. \hat{n} is the unit outward pointing normal. At the i th rod we specify the temperature $T = T_i$. For the temperatures we used $T_b = T_\infty = 1$, $T_1 = 2.0$, $T_2 = 0.1$, $T_3 = 1.5$, and $T_4 = 0.5$. In our test, we used $a = 1$, $b = 1$ and $\varepsilon = 0.1$. The steady state solution is presented in Figure 11. One can clearly see the advantage with this technique. The near-field around the rods is captured and the far-field part is efficiently handled.

6 Conclusions and future work

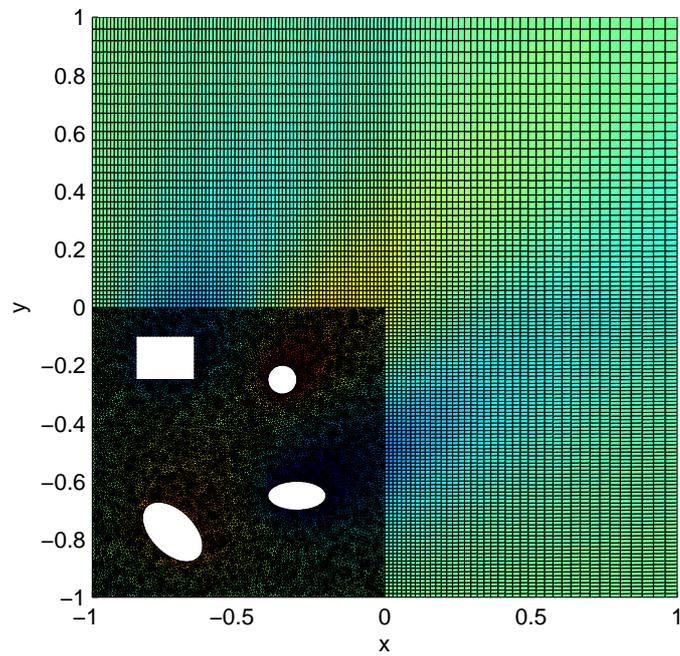
A stable and efficient hybrid method for viscous problems that combines the unstructured finite volume method with the high-order finite difference method has been developed. The hybrid method can be applied to complex geometries with any type of interfaces. The calculations verify that the hybrid method is efficient, accurate and truly stable.

The technique developed in this paper makes it straight forward to apply the hybrid technique to the full Navier-Stokes equations. It also makes it possible to use two existing separate Navier-Stokes solver (one based on UFVM and one on HOFDM) and construct a significantly more efficient hybrid code suitable for aerodynamic and aeroacoustic source to signal type problems.

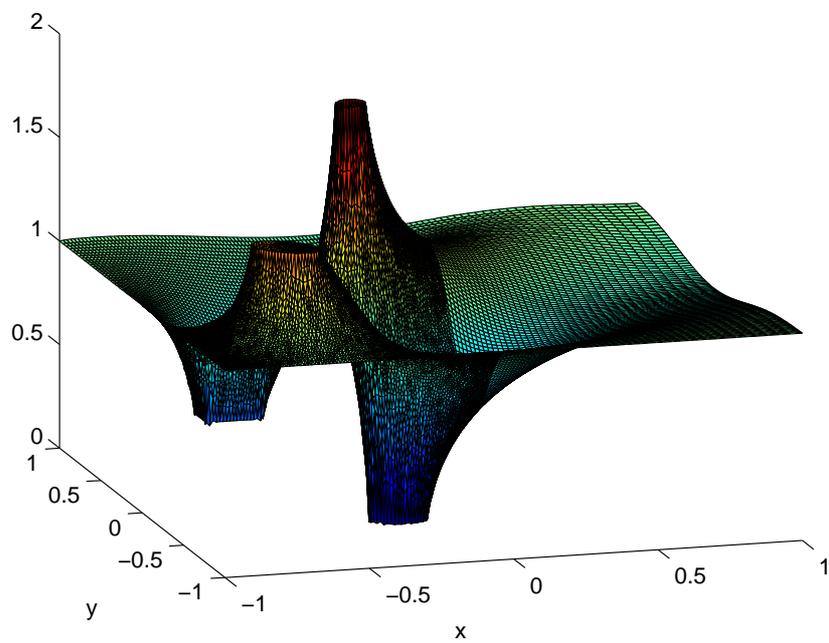
Appendix

Proof: The matrix H in equation (26) can be written in component form,

$$H = \left[\begin{array}{ccc|ccc} a_1 & & & b_1 & & \\ & \ddots & & & \ddots & \\ & & a_n & & & b_n \\ \hline b_1 & & & c_1 & & \\ & \ddots & & & \ddots & \\ & & b_n & & & c_n \end{array} \right] \quad (49)$$



(a) 2D



(b) 3D

Figure 11: The temperature distribution around four rods.

where

$$\begin{aligned}
a_i &= [(P_\xi^R \otimes P_\eta^R) \tilde{J}(\tilde{\xi}_x^2 + \tilde{\xi}_y^2)]_{i,i} > 0, \\
b_i &= [(P_\xi^R \otimes P_\eta^R) \tilde{J}(\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y)]_{i,i}, \quad i = 1, \dots, n \\
c_i &= [(P_\xi^R \otimes P_\eta^R) \tilde{J}(\tilde{\eta}_x^2 + \tilde{\eta}_y^2)]_{i,i} > 0.
\end{aligned} \tag{50}$$

For an arbitrary vector $\mathbf{x} = [x_1, \dots, x_n, y_1, \dots, y_n]^T$, we have

$$\begin{aligned}
\mathbf{x}^T H \mathbf{x} &= \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}^T \begin{bmatrix} a_1 & b_1 \\ b_1 & c_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \dots + \begin{bmatrix} x_i \\ y_i \end{bmatrix}^T \begin{bmatrix} a_i & b_i \\ b_i & c_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \dots + \\
&\quad \begin{bmatrix} x_n \\ y_n \end{bmatrix}^T \begin{bmatrix} a_n & b_n \\ b_n & c_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}
\end{aligned} \tag{51}$$

The eigenvalues of an arbitrary 2×2 matrix on the right-hand-side of (51) is

$$\lambda_{1,2}^i = \frac{a_i + c_i}{2} \pm \sqrt{\left(\frac{a_i + c_i}{2}\right)^2 - (a_i c_i - b_i^2)}, \quad i = 1, \dots, n \tag{52}$$

Since a_i, c_i are positive and

$$\begin{aligned}
a_i c_i - b_i^2 &= [(P_\xi^R \otimes P_\eta^R) \tilde{J}]_{i,i} [(\tilde{\xi}_x^2 + \tilde{\xi}_y^2)_{i,i} (\tilde{\eta}_x^2 + \tilde{\eta}_y^2)_{i,i} - (\tilde{\xi}_x \tilde{\eta}_x + \tilde{\xi}_y \tilde{\eta}_y)_{i,i}^2] \\
&= [(P_\xi^R \otimes P_\eta^R) \tilde{J}]_{i,i} (\tilde{\xi}_y \tilde{\eta}_x - \tilde{\xi}_x \tilde{\eta}_y)_{i,i}^2 > 0,
\end{aligned} \tag{53}$$

$\lambda_{1,2}^i$ is non-negative, which imply that H is a positive semi-definite matrix.

We have proved the lemma. \blacksquare

References

- [1] M.H. Carpenter, D. Gottlieb, and S. Abarbanel. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. *Journal of Computational Physics*, 111(2):220–236, 1994.
- [2] M.H. Carpenter, J. Nordström, and D. Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148:341–365, 1999.
- [3] T. Gerhold, O. Friedrich, and J. Evans. Calculation of complex three-dimensional configurations employing the DLR- τ -code. *AIAA Paper 97-0167*, 1997.
- [4] J. Gong and J. Nordström. Stable, accurate and efficient interface procedures for viscous problems. Technical Report 2006–19, Department of Information Technology, Uppsala University, Uppsala, Sweden, April 2006.

- [5] B. Gustafsson, H.-O. Kreiss, and J. Oliger. *Time Dependent Problems and Difference Methods*. John Wiley & Sons, Inc., 1995.
- [6] A. Haselbacher, J.J. McGuirk, and G.J. Page. Finite volume discretization aspects for viscous flows on mixed unstructured grids. *AIAA Journal*, 37(2), 1999.
- [7] K. Mattsson and J. Nordström. Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics*, 199:503–540, 2004.
- [8] D.J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal*, 28(2), 1990.
- [9] D.J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the Navier–Stokes equations on mixed element meshes. Technical report, Institute for Computer Applications in Science and Engineering, 1995.
- [10] J. Nordström. Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. *Journal of Scientific Computing*, 29(3):375–404, 2006.
- [11] J. Nordström and M. H. Carpenter. Boundary and interface conditions for high order finite difference methods applied to the euler and Navier-Stokes equations. *Journal of Computational Physics*, 148:621–645, 1999.
- [12] J. Nordström and M. H. Carpenter. High-order finite difference methods, multidimensional linear problems and curvilinear coordinates. *Journal of Computational Physics*, 173:149–174, 2001.
- [13] J. Nordström, K. Forsberg, C. Adamsson, and P. Eliasson. Finite volume methods, unstructured meshes and strict stability. *Applied Numerical Mathematics*, 45:453–473, 2003.
- [14] J. Nordström and J. Gong. A stable and efficient hybrid method for aeroacoustic sound generation and propagation. *Comptes Rendus Mecanique*, 333:713–718, 2005.
- [15] J. Nordström and J. Gong. A stable hybrid method for hyperbolic problems. *Journal of Computational Physics*, 212:436–453, 2006.
- [16] B. Strand. Summation by parts for finite difference approximation for d/dx . *Journal of Computational Physics*, 110(1):47–67, 1994.
- [17] M. Svärd and J. Nordström. On the order of accuracy for difference approximations of initial-boundary value problems. *Journal of Computational Physics*, 218(1):333–352, 2006.
- [18] J.M. Weiss, J.P. Maruszewski, and W.A. Smith. Implicit solution of preconditioned Navier–Stokes equations using algebraic multigrid. *AIAA Journal*, 37(1), 1999.