A Comparison of Gradient Approximations for use in Finite Volume Computational Models for Two-Dimensional Diffusion Equations

Jayantha P.A.* and Turner I. W.

School of Mathematical Sciences

Centre in Statistical Science and Industrial Mathematics

Queensland University of Technology,

GPO Box 2434, Brisbane, Australia 4001

May 04, 2001

Abbreviated Title : Finite Volume Methods for Diffusion Equations

Keywords :

heat conduction, isotropic media, cross diffusion term, finite element control volume method, least squares method

*On study leave from the Department of Mathematics, University of Ruhuna, Matara, Sri Lanka

Abstract

Finite volume methods (FVMs) are now a popular choice amongst practitioners in scientific computation and engineering. This paper focuses on generalised FVMs that can be implemented on any mesh structure. The accuracy of FVMs is primarily influenced by the numerical approximation of the flux term at the control volume face. Here, different flux approximations are compared to identify which approximation is the most accurate, independent of the mesh structure. The accuracy of the classical two-node approximation can be improved significantly by using a local gradient reconstruction to capture the cross-diffusion term of the flux at the control volume face. A simple two-dimensional isotropic diffusion equation for which an analytical solution is available is chosen for benchmarking purposes.

Nomenclature

Bi_b Biot numbers

C_p Specific heat $[J/kg/K]$

h Heat transfer coefficient $[W/m^2/K]$

K Thermal conductivity of the material $[W/m/K]$

$\mathbf{n}$ Unit outward normal

S_b Set of boundary points of the solution domain

t Time $[s]$

$\mathbf{t}$ Unit vector perpendicular to $\mathbf{v}$

$\mathbf{u}$ Unit vector along the face

$\mathbf{v}$ Vector representing the distance between two nodes $[m]$

x Coordinate length $[m]$

X Nondimensional length

y Coordinate length $[m]$

Y Nondimensional length

Greek symbols

α Thermal diffiusivity $[m^2/s]$

β_m Eigenvalues from equation (3)

δt Discrete time step size $[s]$

δV_P Area of the control volume $[m^2]$

Θ Nondimensional temperature

μ_n Eigenvalues from equation (4)

ρ Density $[kg/m^3]$

τ Nondimensional time

ϕ Transported quantity - Temperature $[K]$

ϕ_0 Intial temperature $[K]$

ϕ_s Sorrounding temperature $[K]$

Superscripts and Subscripts

b Boundary point

n Represents nth time step

1 INTRODUCTION

Numerous finite volume strategies have been proposed during the last decade to solve a variety of different problems arising in mathematical physics. The control volumes over which the conservation equations have to be integrated are usually polygonal in shape and flux information must be resolved at the midpoint of the control volume faces during the discretisation process. These approximations should be independent of the mesh properties. From the literature it is clear that different mesh structures should be used to treat different geometries and different types of materials. Unstructured control volume methods play a major role in current state of the art computational models.

Turner and Ferguson [1, 2] developed an unstructured mesh cell-centered control volume method that can be applied on meshes consisting of higher order polygonal shapes and that allows complex geometries to be investigated. Decomposition of the flux term into two components was introduced by writing the normal vector ($\hat{\mathbf{n}}_k$) to the corresponding control volume face using the perpendicular vectors $\mathbf{v}_k$, which is the vector that joins two adjacent mesh nodes, and $\hat{\mathbf{t}}_k$. They have discussed a two-node formulation, assuming the cross-diffusion term is not significant when compared to the main term, and a four-node formula, the latter being based on a Hybrid Control Volume Finite Element formulation. The two-node formula was shown to provide satisfactory results only when an orthogonal mesh was used. The idea of orthogonality implies that the angle γ_k, which is the angle between the vector $\mathbf{v}_k$ and the normal vector $\hat{\mathbf{n}}_k$, is zero.

A general cell-centered control volume approach for computational fluid dynamics applications was discussed also by Chow et al.[3] and a two-node flux approximation was used to approximate the diffusion term assuming the cross-diffusion term is zero or near zero compared to the main term if the nonorthogonality of the mesh is not severe.

Turner and Perré [4] again emphasise the necessity of capturing information on the cross diffu-

sion term of the flux to improve the two-node approximation and suggest three two-node gradient approximation methods namely, *orthogonal*, *near orthogonal* and *non-orthogonal*.

A decomposition of the flux term into primary and secondary diffusion fluxes that can be used on unstructured control volume meshes for anisotropic conduction is introduced by Murthy and Mathur [5]. They have used the average of the temperature derivatives of two adjacent cells to construct the secondary diffusion term and the temperature derivatives of cells are obtained by linear reconstruction using the average temperature at the faces. This computation of the secondary diffusion term requires several global iterations for convergence.

Croft [6] also describes the decomposition of the flux at the control volume face, explaining how the accuracy of two node approximation can be improved by introducing a cross diffusion term based on a local reconstruction of the gradient of the control volume face. He also notes that such complicated gradient approximations can be CPU expensive for very large meshes.

A Computational mechanics environment, namely PHYSICA, which facilitates the building of multi-physics models was described by Bailey et al. [7], whereby diffusion terms are approximated again by using a two-node formulation.

Hermeline [8] introduces a coupled finite volume technique defining two meshes called the primary mesh, which consists of finite elements and a dual mesh, which combines centroids of finite elements. The method, which does not depend on mesh regularity, degenerates into the finite difference method on regular grids and has the ability to discretise mixed derivatives terms. However the resulting algorithm requires the solution of a much larger matrix system with many more unknowns than the standard control volume technique. Furthermore, the scheme does not satisfy the maximum principle for non-linear problems unless a constraint is used at every time step to obtain satisfactory results.

Barth [9] introduces a weighted least-squares reconstruction to determine the gradient coefficients

for highly distorted(stretched) meshes, which arise in computational fluid dynamics. This technique is also used by Ollivier-Gooch [10] and Ilinca [11] to estimate derivative components of the function of interest. They highlight the importance of weighting in order to account for the relative distances to neighbours, closer neighbours are more prominent in the gradient approximation than those that are further away from the point of interest.

Extensive strategies and efficient numerical techniques for resolving the sparse matrix systems arising from control volume techniques are discussed by Turner and Perré [4]. They introduce a special data structure to store the resulting sparse matrix and discuss various matrix system solvers and pre-conditioning techniques for use in the *Bi-CGSTAB*, [12], method.

The aim of this work is to evaluate the performance of several different flux approximation techniques for use in a finite volume method that resolves an isotropic heat conduction equation, with emphasis on extending the two-node formulation to incorporate the tangential diffusion component or cross diffusion term into the discretisation. The main objective being to identify which scheme is the most accurate (independent of the mesh structure) efficient, and simplest to implement. To achieve this goal, the following two-dimensional isotropic heat conduction problem for the finite rectangular domain shown in Figure 2d is considered:

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}=\frac{1}{\alpha}\frac{\partial\phi}{\partial t};\;\; 0\le x\le L,\;\; 0\le y\le M;\;\; t>0 \tag{1}$$

where α is the thermal diffusivity and the thermal conductivity can be determined as $K=\alpha\rho C_p$. Generalised boundary conditions are defined on the boundary of the solution domain S_b

$$K\frac{\partial\phi}{\partial n_b}+h_b\phi=h_b\phi_s;\;\; \text{at boundary } S_b,\; t>0 \tag{1b}$$

and initially,

$$\phi(x,y,0)=\phi_0;\quad 0<x<L;\quad 0<y<M. \tag{1c}$$

To construct the control volume mesh, the computational domain is tessellated at first with triangles (or quadrilaterals or a combination of both if required) and the control volumes are constructed around the vertices of these elements by joining the centroids of adjacent cells. This is equivalent to a vertex centred scheme. Note that in some cases, if the centre of the circumcircle is used instead of the centroid for a completely triangular mesh, an orthogonal mesh may result. Such meshes enable efficient two-node approximations for isotropic media (see [1],[3]-[6]). However, placing this restriction on the triangular mesh can sometimes, due to the geometry of the computational domain, cause these centres to move outside of the triangles, whereas the centroid always remains inside.

In this work, both structured and unstructured grids have been used for the discretisation process and a mesh based entirely on triangles was used. In some cases the mesh generator has been forced to generate meshes that have triangular elements that are far from equilateral in order to have strong nonorthogonality of the mesh, thus enforcing a difficult test for the numerical approximations.

Section 3 details the development of the finite volume discretisation process for the diffusion equation, Eq. (1). Although only a first order temporal scheme has been used for the time integration, the time steps used throughout the simulations have been chosen sufficiently small to ensure the accuracy of the spatial scheme is the prominent issue in gauging the overall performance of the numerical solution. Due to the nature of the meshes that were tested, the orthogonality of the mesh is quite poor and the two-node approximation produces misleading and inaccurate results (see also [1, 3, 4, 6]).

Two techniques are introduced to obtain a more accurate approximation of the flux quantities in order to increase the accuracy of the two-node method. Both schemes aim to obtain an accurate expression for the flux at the midpoint of the control volume faces, irrespective of the properties of meshes, as such a strategy ensures second order accuracy of the integration scheme, [13]. However, the resulting schemes require gradients at the mesh node points in order to calculate the cross

diffusion term (or tangential component of the flux). Four different schemes, utilising a local gradient reconstruction have been implemented to obtain the gradient approximations at the mesh vertices. The first three schemes utilise a local gradient reconstruction based on a variant of the divergence theorem, whereby the averaged gradient of the quantity ϕ for the control volume, $(\nabla\phi)_P$, can be obtained by summing the normal weighted ϕ facial values. In the fourth method the gradient for the cell is obtained by using directional derivative information from the node P to all surrounding neighbouring nodes. The resultant over-determined system can be solved using the method of least squares to determine the required gradient components that minimise the square of the norm of the residual of this system(see [11]).

A third technique to approximate the fluxes at the control volume faces uses the hybrid finite volume finite element method, whereby the gradients can be calculated in each triangle using the usual finite element shape functions (see [1, 14]).

This paper is organised as follows. In the next section the theoretical solution of the Eq. (1) is determined. In Section 3 the discretisation of the governing equation is discussed and three techniques to approximate the flux term at the control volume faces of an unstructured control volume mesh are provided. These techniques require gradients to be approximated at the mesh nodes in order to calculate the flux through the faces. Section 4 details four schemes to approximate these gradients. In section 5 the use of Shape Functions to determine the flux through the control volume faces is discussed. In Section 6 a comparison of the numerical solutions against the theoretical solution for different meshes, using different boundary conditions, will be presented. The conclusions of the work are summarised in Section 7.

2 THEORETICAL SOLUTION

Özişik [15] presents exact solutions for both isotropic and anisotropic heat conduction problems. Those methods are used in this work to solve equation (1) analytically. Before finding the exact solution, equation (1) is transformed into dimensionless form by using the following scaling, [1, 15]:

$$X = \frac{x}{L} \quad Y = \frac{y}{M} \quad \tau = \frac{\alpha t}{L^2} \quad Bi_1 = \frac{h_1 M}{K}$$
$$Bi_2 = \frac{h_2 L}{K} \quad Bi_3 = \frac{h_3 M}{K}, \quad Bi_4 = \frac{h_4 L}{K}$$
$$\Theta(X, Y, \tau) = \frac{(\phi - \phi_s)}{(\phi_0 - \phi_s)}; 0 \leq \Theta \leq 1$$

to obtain the non-dimensional equation

$$\frac{\partial^2 \Theta}{\partial X^2} + \frac{L^2}{M^2}\frac{\partial^2 \Theta}{\partial Y^2} = \frac{\partial \Theta}{\partial \tau}; \quad 0 \leq X \leq 1, \quad 0 \leq Y \leq 1, \quad \tau > 0 \tag{2}$$

with boundary conditions

$$\frac{\partial \Theta}{\partial N_b} + Bi_b \Theta = 0 \text{ at boundaries } S_b \text{ for } \tau > 0$$

where N_b, is the outward normal to the computational domain and initially

$$\Theta(X, Y, 0) = 1, \text{ for } 0 < X < 1, \quad 0 < Y < 1.$$

Assuming a separation in the form $\Theta(X, Y, \tau) = \Theta_1(X, \tau)\Theta_2(Y, \tau)$ it is possible to obtain

$$\Theta_1(X, \tau) = \sum_{m=1}^{\infty} \frac{\psi(\beta_m, X)}{N_x(\beta_m)} (\sin \beta_m - \frac{Bi_4}{\beta_m} \cos \beta_m + \frac{Bi_4}{\beta_m}) e^{-\beta_m^2 \tau}$$

where

$$\psi(\beta_m, X) = \beta_m \cos(\beta_m X) + Bi_4 \sin(\beta_m X) \text{ for } 0 < X < 1$$

$$(\beta_m^2 - Bi_2 Bi_4) \tan \beta_m = (Bi_2 + Bi_4)\beta_m; \tag{3}$$

$$N_x(\beta_m) = \frac{1}{2}[(\beta_m^2 + Bi_4^2)(1 + \frac{Bi_2}{\beta_m^2 + Bi_2^2}) + Bi_4]$$

and

$$\Theta_2(Y,\tau) = \sum_{n=1}^{\infty} \frac{\chi(\mu_n, Y)}{N_y(\mu_n)} (\sin\mu_n - \frac{Bi_1}{\mu_n}\cos\mu_n + \frac{Bi_1}{\mu_n}) e^{-\mu_n^2 \tau L^2/M^2}$$

where

$$\chi(\mu_n, Y) = \mu_n \cos(\mu_n Y) + Bi_1 \sin(\mu_n Y) \text{ for } 0 < Y < 1$$

$$(\mu_n^2 - Bi_1 Bi_3)\tan\mu_n = (Bi_1 + Bi_3)\mu_n; \tag{4}$$

$$N_y(\mu_n) = \frac{1}{2}[(\mu_n^2 + Bi_1^2)(1 + \frac{Bi_3}{\mu_n^2 + Bi_3^2}) + Bi_1].$$

The eigenvalues β_m and μ_n are found using a Globally Convergent Newton Method [16]. It was found that some constraints needed to be imposed on the numerical scheme to ensure that none of the important eigenvalue were missed during the root finding process, ensuring that the first 100 eigenvalues for each case always were used in the solution. The analytical solution of the equation (1) can be written as

$$\phi(x,y,t) = \phi_s + \Theta_1(X,\tau)\Theta_2(Y,\tau)(\phi_0 - \phi_s) \text{ for } 0 < x < L, \ 0 < y < M, \ t > 0.$$

3 CONTROL VOLUME TECHNIQUE

A typical control volume within an unstructured mesh of a vertex-centred control volume approach together with the relevant vector descriptions for one control volume face are exhibited in Figure 1a and 1b. In the diagram the control volume shows the centroid, point P, having nodes, N_k, $k = 1, 2, ...p$, ($p = 7$ in this case). The lengths of each face of the control volume are denoted as A_k. The vectors from P to N_k are denoted by $\mathbf{v}_k$. The unit normal vectors to the control volume faces are denoted by $\hat{\mathbf{n}}_k$ and the unit vector along the control volume faces are denoted by $\hat{\mathbf{u}}_k$. Note that the $\hat{\mathbf{u}}_k$'s are traversed in an anti-clockwise direction to ensure that the normal vectors, $\hat{\mathbf{n}}_k$'s, to the control volume faces always point in an outwards direction from the vertex P. The discretised form,

see also [4, 6, 17], of the differential equation (1) is derived by integrating the equation over the control volume (*denoted here as* δV_P) as follows:

$$\int_{\delta V_P} [\frac{\partial \phi}{\partial t} - \alpha \nabla^2 \phi] dV = 0. \tag{5}$$

This equation can be simplified by applying the Divergence theorem in the plane [18]:

$$\frac{d\overline{\phi}}{dt} - \frac{\alpha}{\delta V_P} \oint_{\Gamma_P} \nabla \phi . \hat{\mathbf{n}} d\Gamma \simeq 0; \tag{6}$$

where

$$\overline{\phi} = \frac{1}{\delta V_P} \int_{\delta V_P} \phi dV, \tag{7}$$

is the average of ϕ in a control volume. As there is no approximation made to this point, the Eq. (6) together with (7) is exact, [13].

Assuming that ϕ_P represents the averaged value of ϕ over δV_P and considering the time integral from $n\delta t$ to $(n+1)\delta t$ equation (6) can be written as

$$\frac{\delta V_P}{\delta t}(\phi_P^{(n+1)} - \phi_P^{(n)}) - \alpha \oint_{\Gamma_P} (\nabla \phi)^{(m)} . \hat{\mathbf{n}} d\Gamma \simeq 0. \tag{8}$$

This equation can be written in discrete form as follows:

$$\frac{\delta V_P}{\delta t}(\phi_P^{(n+1)} - \phi_P^{(n)}) - \alpha \sum_{k=1}^{p} (\nabla \phi)_{F_k}^{(m)} . \hat{\mathbf{n}}_k A_k \simeq 0 \tag{9}$$

at interior nodes, P. The line integral in Eq. (8) has been replaced by the sum of the fluxes evaluated at the point F_k, where F_k is the midpoint of face k, multiplied by the corresponding face length. The use of the above assumptions for the discretisation allows a second order approximation in space and first order accuracy in time (see [13]). At the boundary nodes, P, the above equation should be read as

$$\begin{aligned} \frac{\delta V_P}{\delta t}(\phi_P^{(n+1)} - \phi_P^{(n)}) - \alpha \sum_{k=1}^{p} (\nabla \phi)_{F_k}^{(m)} . \hat{\mathbf{n}}_k A_k \\ - \sum_{b=1}^{Nb} h_b (\phi_s - \phi_{F_b}^{(m)}) A_b \alpha / K \simeq 0 \end{aligned} \tag{10}$$

where Nb represents the number of boundary control volume faces. The superscript ^{m} indicates that it is possible to evaluate this term using $m=n+1$ (fully implicit), $m=n$ (fully explicit) or using a value between n and $n+1$ for m, see [17]. The fully implicit scheme, $m=n+1$, will be used for the rest of this work.

3.1 Approximation of flux term

Figure 1b shows a typical control volume face and the following notation will be used throughout the proceeding sections. F_k is the midpoint of the face and the vectors along the line segments R_kF_k, PS_k and M_kN_k (ie. $\delta\mathbf{x}_k$, $\mathbf{s}_k$ and $\mathbf{m}_k$) are perpendicular to the vector $\mathbf{r}_k$., which is the vector from the point S_k to M_k. The unit vectors $\hat{\mathbf{n}}_k$ and $\hat{\mathbf{u}}_k$ are perpendicular to each other and $\mathbf{v}_k$ is perpendicular to $\hat{\mathbf{t}}_k$.

The methods used to approximate the term $(\nabla\phi)^{(m)}_{F_k}.\hat{\mathbf{n}}_k$ in equations (9) and (10) are considered in the next sections.

3.1.1 Two node approximation (Two-Node)

If the angle between the vectors $\mathbf{v}_k$ and $\hat{\mathbf{n}}_k$, γ_k, is approximately zero, (If $\gamma_k=0$, that is the case for an orthogonal mesh), then the term $(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k$ can be replaced by $(\nabla\phi)_{R_k}.\hat{\mathbf{v}_k}$, and hence

$$(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k \simeq (\nabla\phi)_{R_k}.\hat{\mathbf{v}_k} = \frac{\phi_{N_k}-\phi_P}{||\mathbf{v}_k||}. \tag{11}$$

Turner and Ferguson [1] used this *two node formula* and have illustrated that only the results for an unstructured mesh, where the normal vector $\mathbf{n}_k$ and the vector $\mathbf{v}_k$ are approximately coincident, provided reasonable agreement with the results of a rectangular mesh for modelling the wood drying process.

3.1.2 Flux through the midpoint of the face (FM)

Using the directional derivative approximation, or truncated Taylor Series expansion, the flux in the direction of $\hat{\mathbf{n}}_k$ can be written as

$$(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k \simeq \frac{1}{||\mathbf{r}_k||}(\phi_{M_k} - \phi_{S_k}).$$

Similarly, using the same strategy it is possible to write

$$\phi_{M_k} \simeq \phi_{N_k} + (\nabla\phi)_{N_k}.\mathbf{m}_k, \quad \text{and} \quad \phi_{S_k} \simeq \phi_P + (\nabla\phi)_P.\mathbf{s}_k. \tag{12}$$

Thus, a discrete formula for the flux evaluated at the midpoint of the control volume face can be written as follows:

$$(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k \simeq \frac{1}{||\mathbf{r}_k||}\{\phi_{N_k} - \phi_P\} + \frac{1}{||\mathbf{r}_k||}\{(\nabla\phi)_{N_k}.\mathbf{m}_k - (\nabla\phi)_P.\mathbf{s}_k\}. \tag{13}$$

3.1.3 Flux through a representative point of the face (FR)

Another method can be derived by decomposing the vector $\mathbf{v}_k$ in the directions of the orthogonal components $\hat{\mathbf{n}}_k$ and $\hat{\mathbf{u}}_k$, as $\mathbf{v}_k = (\mathbf{v}_k.\hat{\mathbf{n}}_k)\hat{\mathbf{n}}_k + (\mathbf{v}_k.\hat{\mathbf{u}}_k)\hat{\mathbf{u}}_k$. Hence, it follows that

$$(\nabla\phi)_{F_k}.\mathbf{v}_k = (\mathbf{v}_k.\hat{\mathbf{n}}_k)(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k + (\mathbf{v}_k.\hat{\mathbf{u}}_k)(\nabla\phi)_{F_k}.\hat{\mathbf{u}}_k.$$

Note that Turner and Perré [4], and Croft [6] have used the directions $\mathbf{v}_k$ and $\hat{\mathbf{t}}_k$ to decompose $\hat{\mathbf{n}}_k$. Rearranging terms, the following expression is obtained:

$$(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k = \frac{(\nabla\phi)_{F_k}.\mathbf{v}_k}{\mathbf{v}_k.\hat{\mathbf{n}}_k} - \frac{\mathbf{v}_k.\hat{\mathbf{u}}_k}{\mathbf{v}_k.\hat{\mathbf{n}}_k}(\nabla\phi)_{F_k}.\hat{\mathbf{u}}_k.$$

Assuming that the points R_k and F_k are close enough, one can replace $(\nabla\phi)_{F_k}$ by $(\nabla\phi)_{R_k}$. Writing $(\nabla\phi)_{R_k}.\mathbf{v}_k \simeq \phi_{N_k} - \phi_P$ and $(\nabla\phi)_{R_k} \simeq \alpha_k(\nabla\phi)_P + (1-\alpha_k)(\nabla\phi)_{N_k}$ where $\alpha_k PN_k = R_kN_k$ (see Figure 1b), an expression for the flux at the control volume face,

$$(\nabla\phi)_{F_k}.\hat{\mathbf{n}}_k \simeq \frac{\phi_{N_k} - \phi_P}{\mathbf{v}_k.\hat{\mathbf{n}}_k} - \frac{\mathbf{v}_k.\hat{\mathbf{u}}_k}{\mathbf{v}_k.\hat{\mathbf{n}}_k}(\alpha_k(\nabla\phi)_P + (1-\alpha_k)(\nabla\phi)_{N_k}).\hat{\mathbf{u}}_k \tag{14}$$

can be derived . Note that the equations(13) and (14) have the same primary term because $||\mathbf{r}_k|| = \mathbf{v}_k.\hat{\mathbf{n}}_k$ and similar expressions for body-fitted meshes can be found in [5].

4 APPROXIMATION OF GRADIENT

It is clear that expressions (13) and (14) require approximations of the gradient at the mesh nodes in order to be used in the finite volume schemes. The numerical techniques used to provide local approximations are discussed in this section.

For the control volume of interest it is possible to use the Divergence theorem to derive the following identity. Also, refer to the literature, for example, Green-Gauss Reconstruction in [9].

$$\int_{\delta V_P} \nabla\phi dV = \oint_{\Gamma_P} \phi\hat{\mathbf{n}} d\Gamma$$

Using $(\nabla\phi)_P$ as the representative value for $\nabla\phi$ across the control volume and ϕ_{F_k} as the representative value of ϕ on face k (See Figure 1b), and rearranging terms $(\nabla\phi)_P$ can be approximated as

$$(\nabla\phi)_P \simeq \frac{1}{\delta V_P}\sum_{k=1}^{p} \phi_{F_k}\hat{\mathbf{n}}_k A_k. \tag{15}$$

This approximation is also second order as F_k is the midpoint of the control volume face [13]. The following sections describe the methods used to obtain gradients at mesh nodes.

4.1 Gradient using a representative point (GRF)

Assuming that ϕ_{F_k} can be replaced by ϕ_{R_k} and writing $\phi_{R_k} \simeq \alpha_k\phi_P + (1-\alpha_k)\phi_{N_k}$, Eq. (15) can be expressed as

$$(\nabla\phi)_P \simeq \frac{1}{\delta V_P}\sum_{k=1}^{p} \{\alpha_k\phi_P + (1-\alpha_k)\phi_{N_k}\}\hat{\mathbf{n}}_k A_k. \tag{16}$$

It should be noted however that this approximation is not second order. To remedy this problem the following techniques are introduced.

4.2 Gradient using the midpoint for FM (GMF1)

It can be shown that (see Figure 1b)

$$\frac{S_k F_k}{F_k M_k} = \frac{PR_k}{R_k N_k} = \frac{1-\alpha_k}{\alpha_k}.$$

Writing $\phi_{F_k} \simeq \alpha_k \phi_{S_k} + (1-\alpha_k)\phi_{M_k}$ or $\phi_{F_k} \simeq \alpha_k \phi_P + (1-\alpha_k)\phi_{N_k} + \alpha_k (\nabla\phi)_P.\mathbf{s}_k + (1-\alpha_k)(\nabla\phi)_{N_k}.\mathbf{m}_k$, the gradient approximation can be derived as

$$(\nabla\phi)_P \simeq \frac{\Lambda_{P_a}^{-1}}{\delta V_P} \sum_{k=1}^{p} \{\alpha_k \phi_P + (1-\alpha_k)\phi_{N_k} + (1-\alpha_k)(\nabla\phi)_{N_k}.\mathbf{m}_k\} \hat{\mathbf{n}}_k A_k \tag{17}$$

where $\Lambda_{P_a} = [I - \frac{1}{\delta V_P} \sum_{k=1}^{p} \alpha_k A_k \hat{\mathbf{n}}_k (\mathbf{s}_k)^T]$.

4.3 Gradient using the midpoint for FR (GMF2)

Since

$$\nabla\phi_{R_k}.\delta\mathbf{x}_k \simeq \phi_{F_k} - \phi_{R_k}$$

where $\delta\mathbf{x}_k = \overrightarrow{R_k F_k}$, and writing

$$\phi_{R_k} \simeq \alpha_k \phi_P + (1-\alpha_k)\phi_{N_k}$$

and

$$\nabla\phi_{R_k} \simeq \alpha_k (\nabla\phi)_P + (1-\alpha_k)(\nabla\phi)_{N_k}$$

it is clear that

$$\phi_{F_k} \simeq \alpha_k \phi_P + (1-\alpha_k)\phi_{N_k} + (\alpha_k (\nabla\phi)_P + (1-\alpha_k)(\nabla\phi)_{N_k}).\delta\mathbf{x}_k.$$

This is a similar strategy to that proposed by Croft, [6]. Thus, Eq. (15) gives the following gradient approximation

$$(\nabla\phi)_P \simeq \frac{\Lambda_{P_b}^{-1}}{\delta V_P}\sum_{k=1}^{p}\{\alpha_k\phi_P + (1-\alpha_k)\phi_{N_k} + (1-\alpha_k)(\nabla\phi)_{N_k}.\delta\mathbf{x}_k\}\hat{\mathbf{n}}_k A_k \tag{18}$$

where $\Lambda_{P_b} = [I - \frac{1}{\delta V_P}\sum_{k=1}^{p}\alpha_k A_k \hat{\mathbf{n}}_k(\delta\mathbf{x}_k)^T]$.

Some Notes:

(i) The matrices Λ_P evident in both gradient approximations (17) and (18) have elements that are purely geometric quantities that do not vary during the simulation. Consequently Λ_P^{-1} can be generated once the mesh has been read and stored for each control volume. No additional work is required during the simulation.

(ii) Whenever the gradient terms, $\nabla\phi$'s, are approximated using Eqs. (17) and (18) it is suggested to use the latest available gradient information for the term $(\nabla\phi)_{N_k}$. There are two options. Option 1, which is similar to a Gauss-Seidel iterative scheme, uses the latest gradient information at the neighbouring nodes when it is available. Option 2, which is similar to a Jacobi process, uses gradient information only from the previous time level. Both techniques were investigated and found to produce similar results and therefore it is decided to use only the Gauss-Seidel criteria for the simulations performed here.

(iii) Although the gradient approximations derived in Sections 4.2 and 4.3 can be used in either of the flux approximations given by Eqs. (13) and (14) it was decided to use a strategy that is consistent with the way in which the flux expressions were derived. Consequently, (17) was used with (13) and (18) with (14).

4.4 Least-squares gradient reconstruction

Writing $(\nabla\phi)_P.\mathbf{v}_k = \left(\frac{\partial\phi}{\partial x}_P \underline{\mathrm{i}} + \frac{\partial\phi}{\partial y}_P \underline{\mathrm{j}}\right).\left(\Delta x_k \underline{\mathrm{i}} + \Delta y_k \underline{\mathrm{j}}\right) \simeq \phi_{N_k} - \phi_P$ for each node connected to node P, N_k, k=1...p the following over-determined matrix system is obtained (see [9]-[11]).

$$\begin{pmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \cdots & \cdots \\ \cdots & \cdots \\ \Delta x_p & \Delta y_p \end{pmatrix} \begin{pmatrix} \frac{\partial\phi}{\partial x}_P \\ \\ \\ \frac{\partial\phi}{\partial y}_P \end{pmatrix} = \begin{pmatrix} \phi_{N_1} - \phi_P \\ \phi_{N_2} - \phi_P \\ \cdots \\ \cdots \\ \phi_{N_p} - \phi_P \end{pmatrix}$$

or $B_{p\times 2}\Phi'_{2\times 1} = \mathbf{d}_{p\times 1}$. The gradient components that minimise $||B\Phi' - \mathbf{d}||^2$ in the least squares sense with respect to a weighted inner product on $\mathbb{R}^p$ can be determined by multiplying the above system by $W_{p\times p} = Diag(w_k)$ and B^T, to arrive at the normal equations $(B^T W B)_{2\times 2}\Phi'_{2\times 1} = (B^T W \mathbf{d})_{2\times 1}$. This system has the form

$$\begin{pmatrix} \sum_{k=1}^{p} w_k \Delta x_k^2 & \sum_{k=1}^{p} w_k \Delta x_k \Delta y_k \\ \sum_{k=1}^{p} w_k \Delta x_k \Delta y_k & \sum_{k=1}^{p} w_k \Delta y_k^2 \end{pmatrix} \begin{pmatrix} \frac{\partial\phi}{\partial x}_P \\ \\ \\ \frac{\partial\phi}{\partial y}_P \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{p} w_k \Delta x_k (\phi_{N_k} - \phi_P) \\ \sum_{k=1}^{p} w_k \Delta y_k (\phi_{N_k} - \phi_P) \end{pmatrix}. \tag{19}$$

Note that the weight coefficients, w_k's, are chosen so that more importance is given to the directions that are the closest neighbours of the point P as opposed to the nodes are further away from the point P, see [11]. The followings options for c in $w_k = ||\mathbf{v}_k||^{-c}$ were investigated in this work: $c = 0$: LS , $c = 1$: WLSID , and $c = 2$: WLSID2 for all $k = 1...p$, See [9].

5 HYBRID CV-FE FORMULATION

Assuming that ϕ varies linearly across an element, so that $\phi(x, y) = Ax + By + C$, where (x,y) is any point on a triangle and A,B and C are to be determined, and considering the three nodes, see

Figure 1c, which specify any given triangle, the following system of equations can be derived for interpolation purposes:

$$\begin{pmatrix} x_P & y_P & 1 \\ x_{N_k} & y_{N_k} & 1 \\ x_{N_{ka}} & y_{N_{ka}} & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} \phi_P \\ \phi_{N_k} \\ \phi_{N_{ka}} \end{pmatrix}.$$

This system can be solved for A,B and C giving

$$\phi(x,y) = \sum_{r=P,N_k,N_{ka}} M_r \phi_r, \tag{20}$$

where, for example,

$$M_P = \frac{1}{\Omega_a}\{(y_{N_k} - y_{N_{ka}})x + (x_{N_{ka}} - x_{N_k})y + (x_{N_k}y_{N_{ka}} - x_{N_{ka}}y_{N_k})\}$$

and $\Omega_a = x_P(y_{N_k} - y_{N_{ka}}) + x_{N_k}(y_{N_{ka}} - y_P) + x_{N_{ka}}(y_P - y_{N_k})$, where $|\Omega_a|$ is equal to twice of the area of the triangle. Applying the gradient operator, ∇, to Eq. (20) it can be shown that

$$(\nabla\phi)^{(a)} = \sum_{r=P,N_k,N_{ka}} \nabla M_r^{(a)} \phi_r, \tag{21}$$

where superscript $^{(a)}$ indicates the gradient within triangle A. It is easy to show that

$$\sum_{r=P,N_k,N_{ka}} \nabla M_r^{(a)} = 0$$

and hence Eq. (21), after some manipulations and taking the scalar product with the vector $\hat{\mathbf{n}}_k^{(a)}$, can be written as

$$(\nabla\phi)^{(a)}.\hat{\mathbf{n}}_k^{(a)} = \nabla M_{N_k}^{(a)}.\hat{\mathbf{n}}_k^{(a)}(\phi_{N_k} - \phi_P) + \nabla M_{N_{ka}}^{(a)}.\hat{\mathbf{n}}_k^{(a)}(\phi_{N_{ka}} - \phi_P).$$

A similar equation can be written for triangle B as follows:

$$(\nabla\phi)^{(b)}.\hat{\mathbf{n}}_k^{(b)} = \nabla M_{N_k}^{(b)}.\hat{\mathbf{n}}_k^{(b)}(\phi_{N_k} - \phi_P) + \nabla M_{N_{kb}}^{(b)}.\hat{\mathbf{n}}_k^{(b)}(\phi_{N_{kb}} - \phi_P).$$

Therefore the total flux through the face k can be written as

$$\nabla\phi.\hat{\mathbf{n}}_k A_k \simeq (\hat{\mathbf{n}}_k^{(a)})^T G_k^{(a)} \mathbf{d}\phi_k^{(a)} A_k^{(a)} + (\hat{\mathbf{n}}_k^{(b)})^T G_k^{(b)} \mathbf{d}\phi_k^{(b)} A_k^{(b)} \tag{22}$$

where, for example, (see also [19])

$$G_k^{(a)} = \begin{pmatrix} \frac{\partial}{\partial x} M_{N_k}^{(a)} & \frac{\partial}{\partial x} M_{N_{ka}}^{(a)} \\ \\ \frac{\partial}{\partial y} M_{N_k}^{(a)} & \frac{\partial}{\partial y} M_{N_{ka}}^{(a)} \end{pmatrix}, \quad \mathbf{d}\phi_k^{(a)} = \begin{pmatrix} \phi_{N_k} - \phi_P \\ \\ \phi_{N_{ka}} - \phi_P \end{pmatrix} \text{ and}$$
$$\hat{\mathbf{n}}_k^{(a)} = \left(\hat{\mathbf{n}}_{k_x}^{(a)}, \hat{\mathbf{n}}_{k_y}^{(a)}\right)^T. \tag{23}$$

It can be shown that the quantities with superscript $^{(b)}$ can be expressed by replacing a and ka by b and kb in the above equations, respectively. Thus, replacing the term $\nabla\phi.\hat{\mathbf{n}}_k A_k$ in the Eq. (9) by Eq. (22) it follows that

$$\begin{aligned} \delta V_P(\phi_P^{(n+1)} - \phi_P^{(n)}) - \alpha\delta t \sum_{k=1}^{p} \{ & (\hat{\mathbf{n}}_k^{(a)})^T G_k^{(a)} (\mathbf{d}\phi_k^{(a)})^{(n+1)} A_k^{(a)} \\ & + (\hat{\mathbf{n}}_k^{(b)})^T G_k^{(b)} (\mathbf{d}\phi_k^{(b)})^{(n+1)} A_k^{(b)} \} \simeq 0 \end{aligned} \tag{24}$$

where all off diagonal terms have been treated implicitly.

Note:

At the boundary control volumes, the value of the function at the boundary point P is assumed to be the same as that of the boundary surfaces and all discrete quantities are calculated there. If a control volume face coincides with a boundary then the above equations are altered by setting the flux through that face to be equal to the boundary flux, evaluated at point P, multiplied by the length of the boundary control volume face (see [1]). To obtain the cross diffusion terms for the boundary control volumes the gradients at the boundary nodes need to be approximated. These gradients can be approximated using the function values at connected boundary nodes together with boundary flux information.

6 NUMERICAL SIMULATIONS

All of the discretisation procedures discussed in Section 3, 4 and 5 produce a linear system of equations at the vertices P in the following form:

$$a_P \phi_P = \sum_{k=1}^{p} a_k \phi_{N_k} + b_P. \tag{25}$$

For example Eq. (10), after replacing the flux terms by Eq. (13), can be written as

$$\begin{aligned} &\frac{\delta V_P}{\delta t}(\phi_P^{(n+1)} - \phi_P^{(n)}) - \alpha \sum_{k=1}^{p} \frac{1}{||\mathbf{r}_k||}\{\phi_{N_k}^{(n+1)} - \phi_P^{(n+1)}\} \\ &- \alpha \sum_{k=1}^{p} \frac{A_k}{||\mathbf{r}_k||}\{(\nabla\phi)_{N_k}^{(n+1)}.\mathbf{m}_k - (\nabla\phi)_P^{(n+1)}.\mathbf{s}_k\} = \sum_{b=1}^{Nb} \frac{h_b}{K}(\phi_s - \phi_{F_b}^{(n+1)})A_b\alpha \end{aligned} \tag{26}$$

As it is not possible to have the $\nabla\phi$'s at $(n+1)$th time step, $(\nabla\phi)^{(n+1)}$ is replaced by the gradients at nth time step and the function values at boundaries, $\phi_{F_b}^{(n+1)}$'s, are replaced by $\phi_P^{(n+1)}$ to obtain

$$\begin{aligned} &\{\delta V_P + \sum_{k=1}^{p} \frac{A_k}{||\mathbf{r}_k||}\alpha\delta t + \sum_{b=1}^{Nb} \frac{h_b}{K}\alpha\delta t A_b\}\phi_P^{(n+1)} = \sum_{k=1}^{p} \frac{A_k}{||\mathbf{r}_k||}\alpha\delta t\phi_{N_k}^{(n+1)} \\ &+ \delta V_P\phi_P^{(n)} + \sum_{k=1}^{p} \frac{A_k}{||\mathbf{r}_k||}\alpha\delta t\{(\nabla\phi)_{N_k}^{(n)}.\mathbf{m}_k - (\nabla\phi)_P^{(n)}.\mathbf{s}_k\} + \sum_{b=1}^{Nb} \frac{h_b}{K}\alpha\delta t A_b\phi_s \end{aligned} \tag{27}$$

Once all the nodes in the mesh have been visited, a sparse linear system is formed for each of the unknown ϕ_P's. This linear system must be resolved in order to advance the solution in time and an efficient solution strategy must be employed. In this work the resulting linear system is solved using an $ILU(0)$ preconditioned *Bi-CGSTAB* method, storing the sparse matrix in three vectors named *Diagonal*, *Lower* and *Upper*, and two other vectors (namely, *LU_ptrs* that defines the starting positions of the rows in *Lower* and the columns in *Upper*, and *LU_profile* that defines the actual profile of the elements in the sparse matrix) as described by Turner and Perré [4]. Numerical simulation codes, written in C++, were developed to obtain the solutions of the above linear systems.

To test the flux approximations discussed throughout Section 3, together with relevant gradient approximations discussed in Section 4, a wood like material is used. The two-dimensional computational domains are illustrated in Figure 2. Three different domains are selected as described by

$ABCD$ in Figures 2a, 2b and 2c. In **Case 1** the full domain, Figure 2a, is considered. In this case all faces are subjected to the mixed boundary conditions. Due to the symmetry of this problem and to invoke a different set of boundary conditions on the benchmark problem, only a quarter of the computational domain is used for **Case 2** and **Case 3** , see Figures 2b and 2c. In this case the symmetry planes have the flux set to zero.

Throughout the tests a material with the physical properties shown in Table 1A is used with $\phi_0 = 30^0C$ and $\phi_s = 140^0C$. Different time steps used for the tests and found all the shemes converge without any significant changes in the solutions. The time step, δt, used for the simulations presented here is one second and the results shown are obtained after 1000 time steps. The meshes used throughout the analysis are shown in Figures 3 and 6. These meshes were created using the mesh generator, EasyMesh, developed by Niceno [20]. However the centroids of the triangles are used as element centres in the construction of the control volumes. The five meshes include Mesh 1 (Case 1) - a uniform mesh in the domain $0.1 \times 0.04m^2$ and Mesh 2 (Case 1) - an unstructured mesh in the domain $0.1 \times 0.04m^2$, Mesh 3 (Case 2) - a middle coarse mesh in the domain $0.05 \times 0.02m^2$ and Mesh 4 (Case 2) - an unstructured mesh in the domain $0.05 \times 0.02m^2$, and Mesh 5 (Case 3) - an unstructured mesh in the domain $0.5 \times 0.02m^2$.

In order to gauge the accuracy of each numerical scheme in comparison with the exact solution the errors are calculated using the following Root Mean Squared Error formula:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{M_n} \left(\phi_{k,exact} - \phi_k\right)^2}{\sum_{k=1}^{M_n} \left(\phi_{k,exact}\right)^2}} \times 100 \tag{28}$$

where ϕ_k is the value of the variable at point k, $\phi_{k,exact}$ is the exact solution at the same point, and M_n is number of points. Table 1 summarises completely all the findings of this work. Note that the acronyms used in the table coincide with the section headings given throughout Sections 3, 4 and

5. Hence, the scheme FR-GRF, for example, implies flux evaluated at a representative point on the control volume face, Eq. (14), with cross diffusion term computed according to Eq. (16).

The results for some selected simulations have been exhibited in Figure 4 for Mesh 2 and Case 1, Figure 5 for Mesh 4 and Case 2 and Figure 6 for Mesh 5 and Case 3. Note that in all graphs the exact solution is plotted as a flood contour and the relevant finite volume approximation methods are plotted as solid lines. Furthermore, in some cases due to the poor quality of the meshes used for the computations and the interpolation technique used by the plotting software, the true symmetry of the solution is slightly concealed. Since the results for the near orthogonal meshes (Figure 3a and 3c) exhibited good agreement with the exact solution it was decided not to include these figures in this text. The interested reader can find this detailed information in [21].

The first observation to be made from the Table 1B and the supporting graphs is that the two-node approximation can produce misleading and inaccurate results. For example, for Meshes 2, 4 and 5, the RMSEs are approximately 3.9%, 5% and 1.6% respectively. These errors are a factor of 4, 10 and 3 times larger than all other schemes investigated for the respective meshes. In fact, the Figures 4- 6 reflect this error, particularly for the completely unstructured Meshes 2, 4 and 5, where the two-node approximation provides results that are far from the exact solution.

Furthermore, it is possible to observe from Table 1B that FR performs better than FM in nearly all of the numerical simulations performed. One possible explanation for this finding is that more approximations are necessary in FM in comparison with FR. For example, FM requires ϕ_M and ϕ_S to be approximated (see Eq. 12), whereas, in FR, the only approximation that is used is that the point R_k is sufficiently close to the midpoint of the CV face F_k. Note, however, that for Mesh 2, FM-WLSID2 did outperform FR-WLSID2.

Overall, the method WLSID2 with either FM or FR for a generalised finite volume approach is recommended, since this combination provides the smallest RMSE for all of the tests carried

out here. However, LS and WLSID did provide smaller errors than WLSID2 with FR for Mesh 4. The next best overall performer was the hybrid CV-FE formulation, however, as noted in the text, this method depends on the shapes of the elements and is not therefore a generalised (mesh independent) FV technique. Note for mesh 4, FR-GMF2 outperforms all other schemes and for Mesh 2, FM-WLSID2 is the best.

The least expensive in terms of CPU cost are the GRF schemes for the generalised FV methods and the hybrid technique for the element shape dependent FV schemes. These methods also produce reasonable RMSEs (again refer to Table 1B). In particularly, it is possible to see that GRF can correct the two-node approximation competitively with all other methods. In most cases GMF1 and GMF2 do perform better than GRF, however, the penalty paid here is the additional computational overheads associated with providing a second order gradient approximation such as the one proposed in Equations (17) and (18). The more expensive schemes, in terms of CPU and additional storage requirements, would be the least squares methods. In fact, the LS methods have additional computational costs associated with them in comparison with the other methods, due to the solution of the normal equations for each node in the mesh for every time step.

If storage is an issue, the scheme that requires the least amount of data storage (i.e. additional gradient information) is the Hybrid method followed by GRF. The GRF method requires no additional storage, but at each time step the gradients for every node point must be determined via equation (16). The next least expensive in terms of memory would be either GMF1 or GMF2, which require the geometrical matrix Λ_P^{-1} to be computed and stored at the first pass through the mesh. The LS schemes also requires a matrix of geometrical properties to be stored for each node. Recall, however, that these matrices are computed and inverted only once at the program initialisation stage once the mesh has been read.

In terms of program implementation (Coding, data structures, etc) GRF is the easiest, followed

by the hybrid method and then WLS and finally GMF. For structured meshes (Mesh 1 and Mesh 3) it was found that the best scheme in terms of accuracy was the FR flux approximation with any of the gradient approximations. In fact, GRF would be sufficient in this case. For unstructured meshes (Mesh 2, Mesh 4 and Mesh 5), the FR flux approximation appears the best, in combination with any of the LS methods.

In Table 2 the computational requirements, in terms of the total number of BiCGSTAB iterations with ILU(0) preconditioner and CPU time, are exhibited for all flux approximation methods for Mesh 4 and Mesh 5. From this table it is possible to observe that BiCGSTAB always converged in approximately the same number of iterations for all FM and FR flux approximation methods. However, the hybrid method required approximately 1950 and 975 additional iterations for Mesh 4 and Mesh 5 respectively, because of the reduction in diagonal dominance of the system matrix due to the implicit treatment of the off diagonal nodes. All FR and FM schemes required approximately the same CPU time for the simulation and the hybrid method was the fastest scheme.

7 CONCLUSION

This paper has presented three different flux approximation schemes for use in a finite volume method for discretising a two-dimensional isotropic diffusion equation. It was found that the accuracy of a two-node approximation was quite poor for the meshes tested here, since these meshes did not in general satisfy the orthogonality constraint and consequently, the cross diffusion term is important and must be used to correct this error. Five different methods for computing this term were discussed. In summary, if the mesh element shape is consistent with the hybrid CV-FE methodology, this scheme is highly recommended. It performs well for either a structured or unstructured mesh. Furthermore, the hybrid method is simple to implement and is the fastest in terms of CPU time for

the tests performed here. For a completely generalised FV method, FR with any of the LS schemes is recommended for accuracy, however, these methods can be CPU intensive. If speed is the primary concern, FR or FM with GRF is more than adequate. In all cases, it must be recognised that the two-node approximation provides poor accuracy for most unstructured meshes and therefore must be corrected using any of the schemes discussed in this paper for capturing the cross diffusion term. These concepts will be extended to three-dimensional cases considering much more skewed meshes than the ones examined here, and this will form the basis of continuing work on diffusion in strongly anisotropic and heterogeneous media.

Acknowledgements

The first author wishes to acknowledge the financial support provided by the Queensland University of Technology under the IPRS scholarship program and the leave granted for the studies from the University of Ruhuna, Sri Lanka, where he is affiliated as a Lecturer. Both authors would like to thank the referees for their constructive comments on improving the overall presentation of this paper.

References

[1] I. W. Turner and W. J. Ferguson, An unstructured mesh cell-centered control volume method for simulating heat and mass transfer in porous media: Application to softwood drying, Part I: The isotropic model, Appl. Math. Modelling, Vol. 19, pp. 654-667, 1995.

[2] I. W. Turner and W. J. Ferguson, An unstructured mesh cell-centered control volume method for simulating heat and mass transfer in porous media: Application to softwood drying, Part II: The anisotropic model, Appl. Math. Modelling, Vol. 19, pp. 668-674, 1995.

[3] P. Chow, M. Cross, and K. Pericleous, A Natural Extension of the Conventional Finite Volume

Method into Polygonal Unstructured Meshes for CFD Application, Appl. Math. Modelling, Vol. 20, pp. 170-183, 1996.

[4] I. Turner and P. Perré, A Synopsis of the Strategies and Efficient Resolution Techniques used for Modelling and Numerically Simulating the Drying Process, in I. Turner A. S. Mujumdar (eds.), Mathematical Modeling and Numerical Techniques in Drying Technology, Chap. 1, Marcel Dekker, Inc. New York, 1996

[5] J. Y. Murthy and S. R. Mathur, Computation of Anisotropic Conduction using Unstructured Meshes, Journal of heat Transfer, Vol. 120, pp. 583-591, 1998.

[6] N. Croft, Unstructured Mesh - Finite Volume Algorithms for Swirling, Turbulent, Reacting Flows, PhD. Thesis, University of Greenwich, London, UK, 1998.

[7] C. Bailey, G, A. Taylor, M. Cross and P. Chow, Dicretisation Procedures for multi-Physics Phenomena, Journal of Computational and Applied Mathematics, Vol. 103, pp. 3-17, 1999.

[8] F. Hermeline, A Finite Volume Method for the Approximation of Diffusion Operators on Distorted Meshes, Journal of Computational Physics, Vol. 160, pp. 481-499, 2000.

[9] T. J. Barth, Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations, in: Lecture Notes Presented at the VKI Lecture Series 1994-05, Feb. 1994.

[10] C. F. Ollivier-Gooch, A New Class of ENO Schemes Based on Unlimited Data-Dependent Least-Squares Reconstruction, in: AIAA - 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, US, AIAA-96-0887, 1996.

[11] C. Ilinca, X. D. Zhang, J. -Y. Trépanier and R. Camarero, A Comparison of Three Error Estimation Techniques for Finite-Volume Solutions of Compressible Flows, Computer Methods in Applied Mechanics and Engineering, Vol. 189, pp. 1277-1294, 2000.

[12] H.A. Van Der Vorst, BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solutions of Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., Vol 13, No. 2, pp. 631-644, 1992.

[13] E. Turkel, Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows, ICASE Report No. 85-43, NASA, Virginia, 1985.

[14] W. J. Ferguson, The Control Volume Finite Element Numerical Solution Technique Applied to Creep in Softwoods, Int. J. Solids Structures, Vol. 35, No.13, pp.1325-1338, 1998.

[15] M. N. Özişik, Heat Conduction, 1st ed., Wiley, New York, 1980.

[16] A. Jennings, J. J. McKeown, Matrix Computation, 2nd ed., John Wiley & Sons Ltd, New York, 1992.

[17] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, 1980.

[18] J. E. Marsden and A. J. Tromba, Vector Calculus, 4th ed., pp.466-476, W. H. Freeman and Company, 1998.

[19] P. Perré and I. Turner, TransPore: A Generic Heat and Mass Transfer Computational Model for Understanding and Visualising The Drying Of Porous Media, Invited paper, Drying Technology Journal, Vol. 17(7), 1273-1289, 1999.

[20] B. Niceno, "EasyMesh" (Version 1.4), freely available mesh generator on the web Site: `http://www-dinma.univ.trieste.it/~nirftc/research/easymesh/`.

[21] P. A. Jayantha, PhD Thesis (to be submitted), Queensland University of Technology, Brisbane, Australia, 2002.

Table 1: The physical values used in this work and Comparison of RMSEs

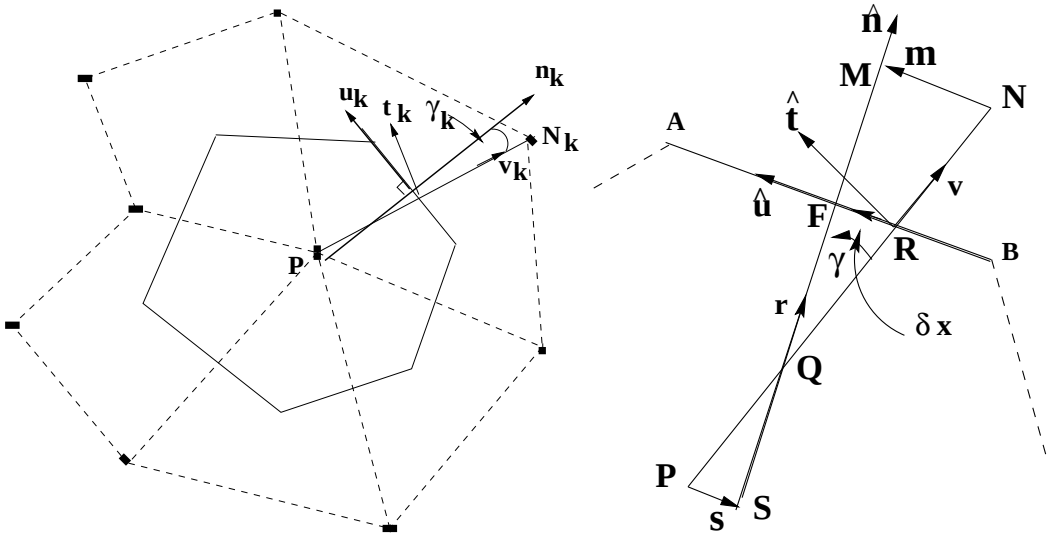
(**A**)	L	M	K	α	h_1 at	h_2 at	h_3 at	h_4 at
				$\times 10^7$	$y = 0$	$x = L$	$y = M$	$x = 0$
Units	m	m	$W/m/K$	m^2/s		$W/m^2/K$		
Case 1	0.1	0.04	0.154	1.52	10	10	10	10
Case 2	0.05	0.02	0.154	1.52	0	10	10	0
Case 3	0.5	0.02	0.154	1.52	0	10	10	0

(**B**)	Gradient Approx.	Mesh 1 (Case 1)	Mesh 2 (Case 1)	Mesh 3 (Case 2)	Mesh 4 (Case 2)	Mesh 5 (Case 3)
Two-Node		0.217234	3.894251	0.402431	5.034711	1.634527
FM	GRF	0.079019	0.875013	0.066320	0.571933	0.501953
	GMF1	0.080157	0.661019	0.065465	0.459624	0.510724
	LS	0.080050	0.908007	0.066594	0.508964	0.515209
	WLSID	0.080121	0.585588	0.064633	0.409906	0.483765
	WLSID2	0.080178	0.303930	0.062810	0.418755	0.459563
Hybrid		0.072877	0.316189	0.069594	0.439017	0.403097
FR	GRF	0.072465	0.791738	0.057568	0.399316	0.383347
	GMF2	0.072406	0.749109	0.057657	0.319736	0.384828
	LS	0.072297	0.804576	0.058742	0.372446	0.387500
	WLSID	0.072336	0.553889	0.056869	0.353115	0.372264
	WLSID2	0.072401	0.322934	0.055113	0.407310	0.362414

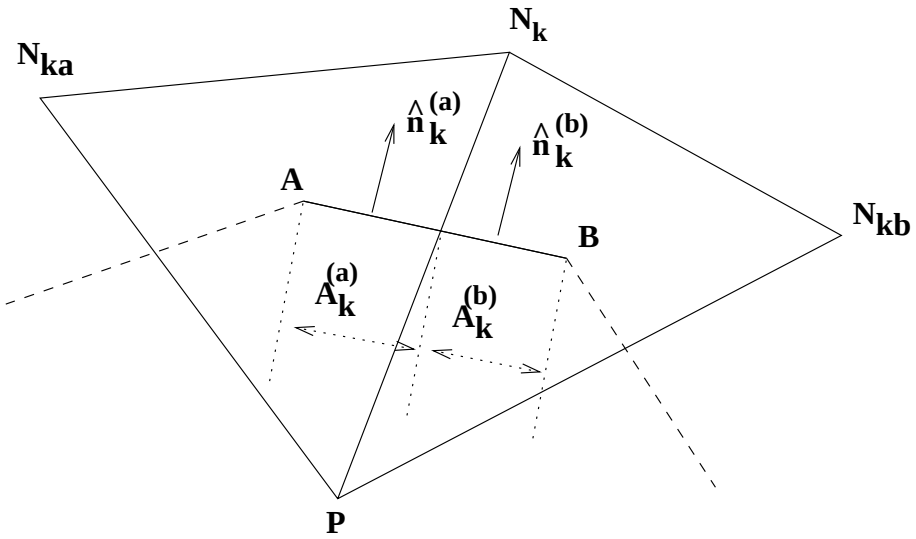
Table 2: Computation time and total No. of Bi-CGSTAB+ILU(0) iterations

(On a Pentium III 450 MHz PC with 256 Mb of RAM)

Flux Approximation	Number of Iterations (Mesh 4)	CPU time $\times 1000s$ (Mesh 4)	Number of Iterations (Mesh 5)	CPU time $\times 1000s$ (Mesh 5)
FR-GRF	4983	1842	2066	3705
FM-GRF	4987	1732	2085	3375
FR-GMF2	4990	1852	2066	3955
FM-GMF1	4999	1802	2086	3605
FR-LS	4991	1802	2068	3715
FM-LS	4990	1782	2082	3314
FR-WLSID	4984	1842	2060	3935
FM-WLSID	4988	1792	2079	3585
FR-WLSID2	4974	1862	2052	3935
FM-WLSID2	4987	1762	2075	3545
Hybrid	6933	1492	3049	1802

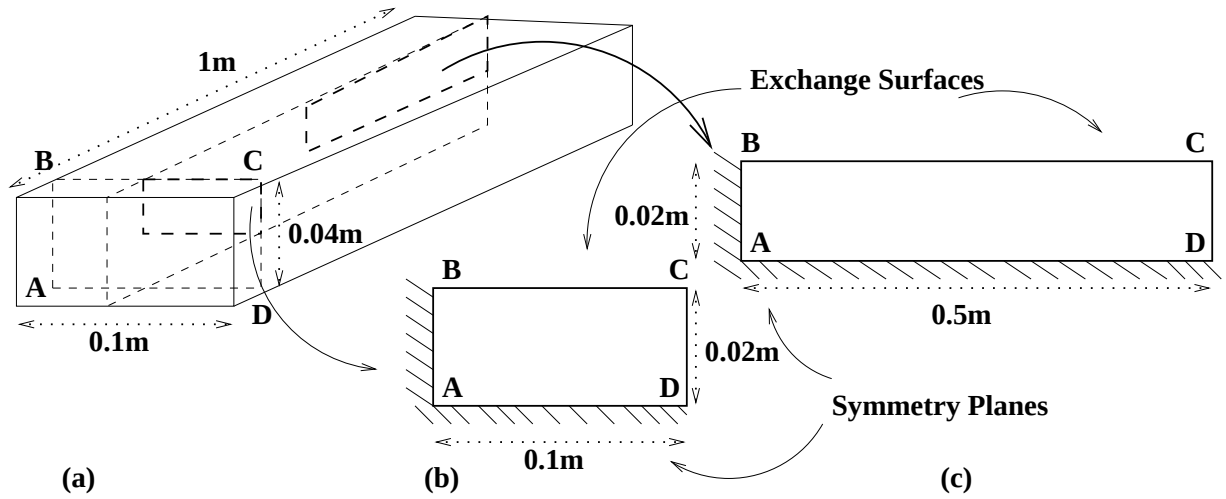


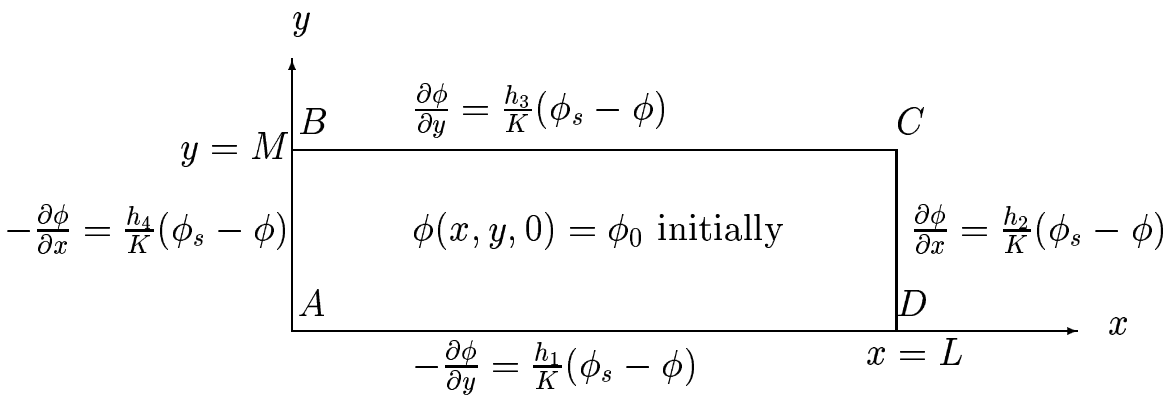
(a) (b)



(c)

Figure 1: (a) A control volume of an unstructured Mesh, (b) A representative control volume face, (c) Two Adjacent triangles of a control volume finite element mesh





(d)

Figure 2: (a)-(c) Physical configuration and computational domains, (d) Boundary and initial conditions for a rectangular domain

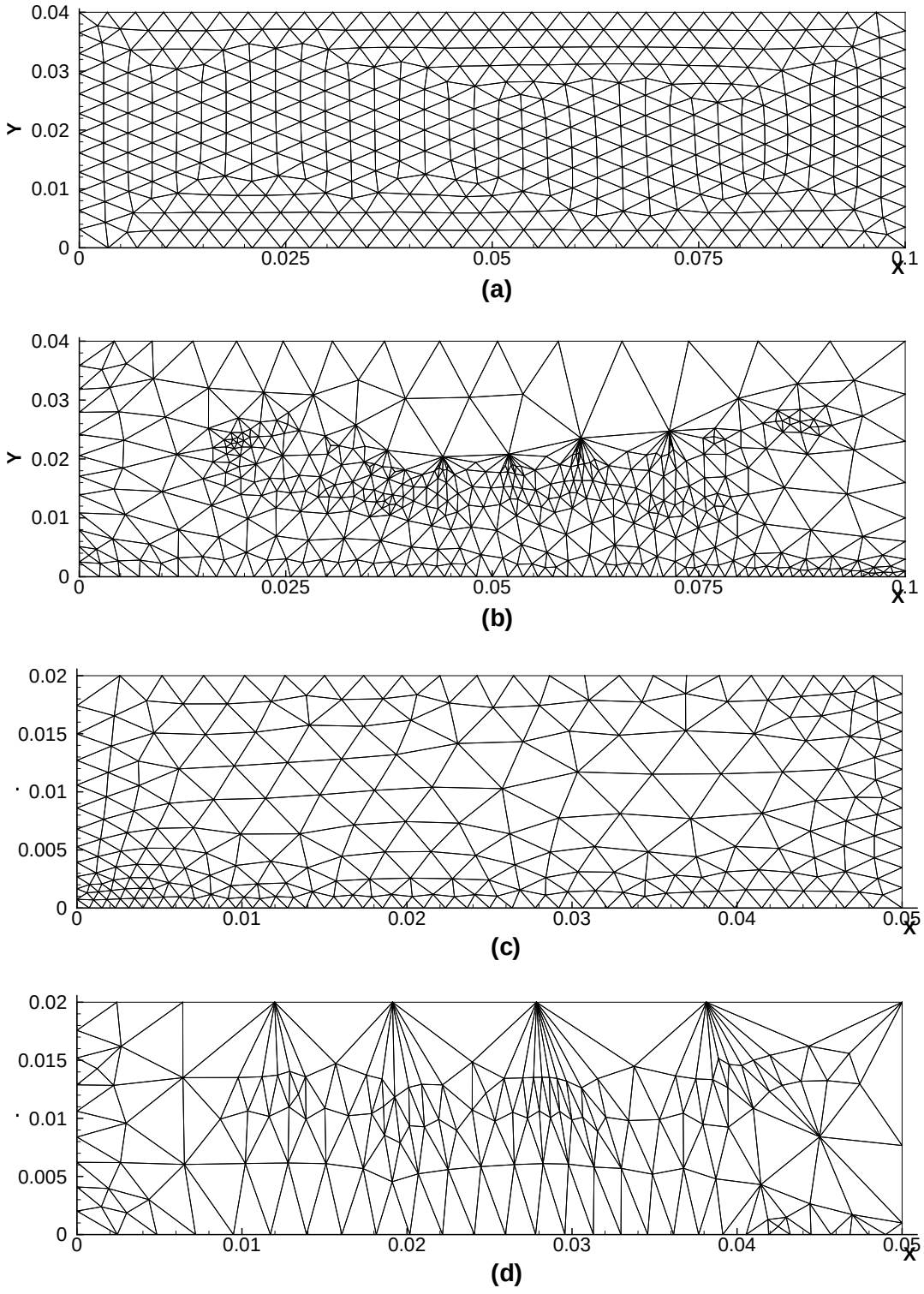


Figure 3: Computational Meshes used for all Simulations (a) **Mesh 1**- Structured mesh($0.1m \times 0.04m$), (b)**Mesh 2** - Unstructured mesh ($0.1m \times 0.04m$) (c) **Mesh 3** - Middle coarse mesh ($0.05m \times 0.02m$) and (d) **Mesh 4** - Unstructured mesh ($0.05m \times 0.02m$)

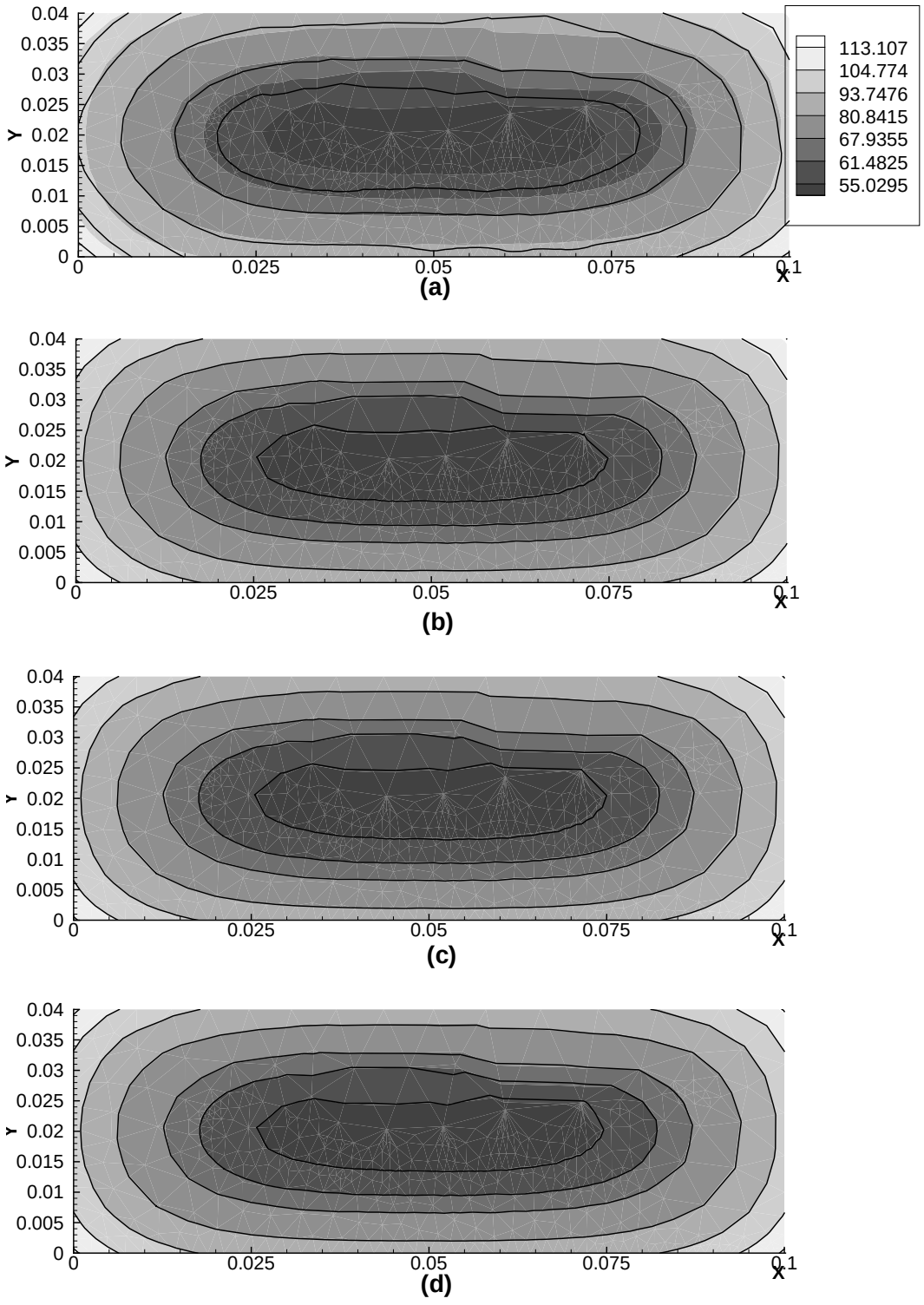


Figure 4: Comparison of analytical(flood contours) and numerical solutions(lines) on Mesh 2 for Case 1 using (a) Two-Node approximation (b) Scheme FM with WLSID2 (c) Scheme FR with WLSID2 and (d) Hybrid finite volume finite element method.

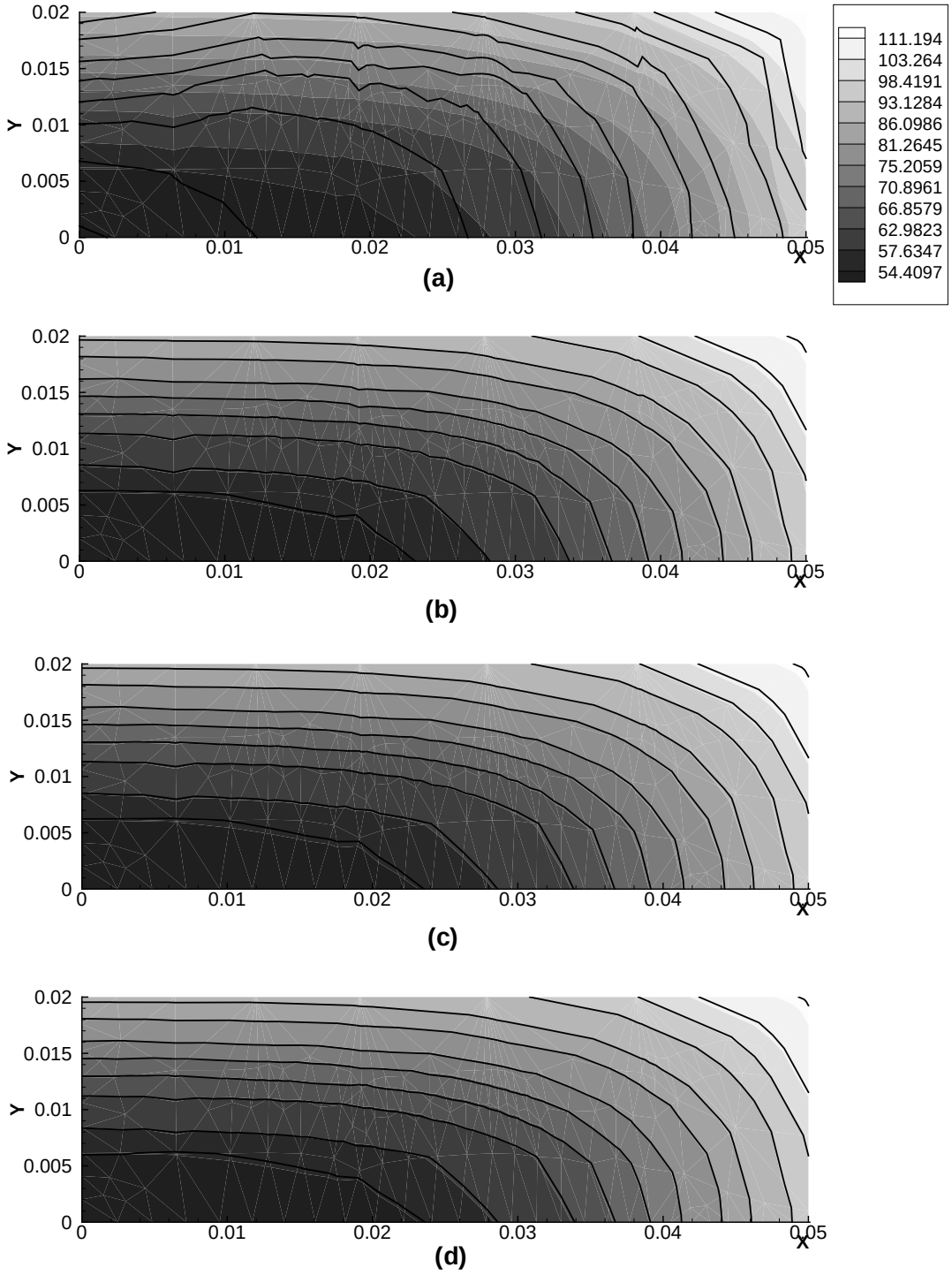


Figure 5: Comparison of analytical(flood contours) and numerical solutions(lines) on Mesh 4 for Case 2 using (a) Two-Node approximation (b) Scheme FM with WLSID (c) Scheme FR with WLSID and (d) Hybrid finite volume finite element method.

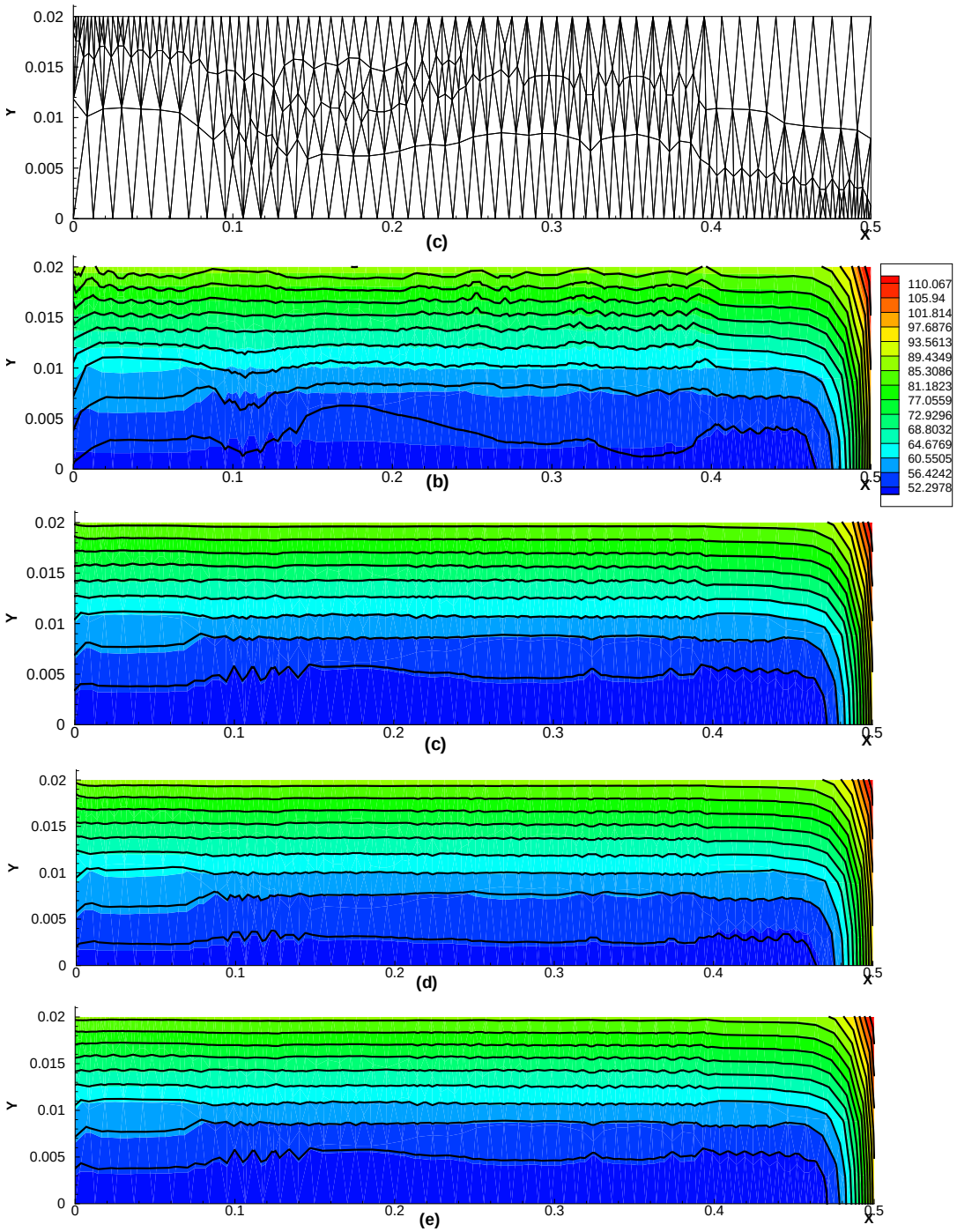


Figure 6: Comparison of analytical(flood contours) and numerical solutions(lines) on (a) **Mesh 5** for Case 3 using (b) Two-Node approximation (c) Scheme FR with WLSID2 (d) Scheme FM with WLSID2, (e) Hybrid finite volume finite element method.