JNC TN9400 2000-094

Discretization of Convection-Diffusion Equations with Finite-Difference Scheme Derived from Simplified Analytical Solutions

September, 2000

O-arai Engineering Center
Japan Nuclear Cycle Development Institute

JNC TN9400 2000-094

September, 2000

DISCRETIZATION OF CONVECTION-DIFFUSION EQUATIONS WITH FINITE-DIFFERENCE SCHEME DERIVED FROM SIMPLIFIED ANALYTICAL SOLUTIONS

Vladimir Kriventsev*

ABSTRACT

Most of thermal hydraulic processes in nuclear engineering can be described by general convection-diffusion equations that are often can be simulated numerically with finite-difference method (FDM). An effective scheme for finite-difference discretization of such equations is presented in this report. The derivation of this scheme is based on analytical solutions of a simplified one-dimensional equation written for every control volume of the finite-difference mesh. These analytical solutions are constructed using linearized representations of both diffusion coefficient and source term. As a result, the Efficient Finite-Differencing (EFD) scheme makes it possible to significantly improve the accuracy of numerical method even using mesh systems with fewer grid nodes that, in turn, allows to speed-up numerical simulation.

EFD has been carefully verified on the series of sample problems for which either analytical or very precise numerical solutions can be found. EFD has been compared with other popular FDM schemes including novel, accurate (as well as sophisticated) methods. Among the methods compared were well-known central difference scheme, upwind scheme, exponential differencing and hybrid schemes of Spalding. Also, newly developed finite-difference schemes, such as the quadratic upstream (QUICK) scheme of Leonard, the locally analytic differencing (LOAD) scheme of Wong and Raithby, the flux-spline scheme proposed by Varejago and Patankar as well as the latest LENS discretization of Sakai have been compared. Detailed results of this comparison are given in this report. These tests have shown a high efficiency of the EFD scheme. For most of sample problems considered EFD has demonstrated the numerical error that appeared to be in orders of magnitude lower than that of other discretization methods. Or, in other words, EFD has predicted numerical solution with the same given numerical error but using much fewer grid nodes.

In this report, the detailed description of EFD is given. It includes basic assumptions, the detailed derivation, the verification procedure, as well verification and comparisons. Conclusion summarizes results and highlights the problems to be solved.

*) JNC International Fellow

Thermal Hydraulics Group, System Engineering Technology Division, O-arai Engineering Center, JNC, Japan

JNC TN9400 2000-094
2000年9月

簡易解に基づく有限差分法による移流ー拡散方程式の離散化

(研究報告)

Vladimir Kriventsev*

要旨

原子力分野における伝熱流動現象の多くは、一般的な移流ー拡散方程式で記述でき、ほとんどの場合は、有限分差分法によって数値的に解くことができる。本報告では、移流ー拡散方程式の差分化に有効なスキームについて述べる。本スキームは、有限差分メッシュの全てのコントロールボリュームに関する一次元の簡略化した方程式の解析解に基づいて導出されている。解析解は、拡散項とソース項を線形化して求めた。こうして導出された実効的差分スキーム(EFD)を用いれば、少ないメッシュ数でも解析精度を大幅に向上させることが可能であり、その結果、計算時間を節約できる。

ここで開発したEFDスキームの検証のため、解析解が既知の問題や、詳細な数値解が求められている問題を解析した。EFD スキームと他の一般的に使われている差分スキームとも比較した。例えば、中心差分スキーム、風上差分、Spaldingによる指数差分とハイブリッド差分と比較した。また、比較的最近に提案されたLeonardのQUICKスキーム、WongとRaithbyのLOADスキーム、VaewjagoとPatankarのフラックス=スプラインスキーム、酒井のLENSスキームとの比較も行った。本報告書にはそれらの詳細な比較を述べる。本研究で提案するEFDスキームは、他の方法と比べて多くの問題で数桁ほど解析精度が良いこと、あるいは同じ解析精度を得るために少ないメッシュしか必要としないことが示された。

本報告書には、EFD スキームに関して詳細に記述する。基本的な仮定、導出過程、検証手順、検証解析と比較に関して述べた後、結論には結果のまとめと今後の課題を述べる。

*大洗工学センター　システム技術開発部　熱流体技術開発グループ、国際特別研究員

CONTENTS

LIST OF FIGURES

NOMENCLATURE

$a_0, a_1, a_2 \ldots$	polynomial coefficients
$a_i, b_i, c_i, d_i \ldots$	coefficients of finite-difference discretization
c_p	heat capacity
F	surface of the control volume
f	transported quantity
J	flux of transported quantity
$\tilde{K}$	diffusion-correction factor
L	length of one-dimensional calculation area
N_{points}	number of mesh points
P	absorption term coefficient
Pe	mesh Peclet number, $\frac{\rho u \Delta x}{\Gamma}$
Pr	Prandtl number, $\frac{\rho c_p \nu}{\lambda}$
Pr^T	turbulent Prandtl number, $\frac{\rho c_p \nu^T}{\lambda^T}$
Q	physical source term in transport equation
S	aggregate source term contributed by term from other directions
$\tilde{S}$	an extra source term in finite-difference discretization
T	temperature
$\vec{u}, u_x, u_y, u_z$	velocity vector and its components in regular Cartesian coordinate system
u_*	dynamic velocity, $\sqrt{\zeta / 8}$
V	volume of control volume (area in two-dimensional case)
w_1, w_2	velocity components in rotated coordinate system
x,y,z	Cartesian coordinates
y^+	non-dimensional distance to the wall, $y^+ = y u_* / \nu$

	Greek Symbols
ε	numerical accuracy
Δ	indicates an increment of value
Γ	diffusion coefficient
λ	thermal conductivity
λ^T	turbulent thermal conductivity
ν	viscosity
ν^T	turbulent viscosity
ρ	density
τ	time
ζ	friction factor

1. INTRODUCTION

Most transfer processes including fluid flow and heat transfer can be described by general convection-diffusion conservation equation:

$$\frac{\partial(\rho f)}{\partial \tau} + \mathrm{div}(\rho \vec{u} f) = \mathrm{div}(\Gamma \vec{\nabla} f) + S, \qquad (1)$$

where f is a scalar value that is convected by a flow moving with velocity $\vec{u}$ and density ρ and diffuses with diffusion coefficient Γ. S is the volumetric rate of f-generation.

Frequently, for many engineering problems, equation (1) should be solved numerically in three dimensions in order to predict velocity and temperature distribution correctly. Advanced turbulence models, such as Large Eddy Simulation (LES), also demands transient simulation in three-dimensions. Whereas this kind of analysis requires a lot of CPU time, the great attention should be paid to the numerical method and discretization technique.

In many situations, Finite-Difference Method (FDM) is a choice. Numerical procedure of FDM consists of the following main steps:

i) Dividing of calculation area in elementary control volumes [1] using orthogonal or non-orthogonal mesh system

ii) Discretization of Eq. (1) on control volumes that results in a linear equation for every control volume

iii) Solving of the system of these linear equation

In this report, we will concentrate on discretization of Eq. (1) over orthogonal mesh systems. Along with extensively used conventional discretization procedures for solving Eq. (1), such as a central difference scheme, an upwind scheme, exponential differencing and hybrid schemes of Spalding and a power-law scheme of Patankar [1], a number of new formulations have been developed in recent years. Above all else, mention should be made of the quadratic upstream (QUICK) scheme of Leonard [2], the locally analytic differencing (LOAD) scheme of Wong and Raithby [3], the flux-spline scheme proposed by Varejago in conjunction with the multigrid solution method [4], as well as the latest LENS discretization of Sakai [5]. Above mentioned new formulations enable more accurate solution using a mesh system with fewer grid nodes. Most of them use one-dimensional locally "exact" analytical solution in discretization, which helps to increase the precision dramatically. However, at the same time, they require more computational efforts and sometimes may result in oscillatory and even divergent solutions. Further improvement of locally analytical approach while taking into account spatial distribution of diffusion coefficient Γ was proposed [7,8]. This work reports the improved efficient finite-differencing (EFD) scheme with extension on transient convection-diffusion problems of fluid flow and heat transfer. The detailed EFD formulation is also given.

2. THE EFFICIENT FINITE-DIFFERENCE (EFD) SCHEME DERIVATION

Common EFD procedure is based on the standard control-volume method [1]. In general, in order to obtain a finite-difference form of the differential equation, we integrate Eq. (1) over the control volume according to Patankar [1]. This integration gives the following

$$\frac{\partial \iiint_V \rho f dV}{\partial \tau} + \iint_F \left(\rho \vec{u} f - \Gamma \vec{\nabla} f\right) d\vec{F} = \iiint_V S dV . \tag{2}$$

The finite-difference discretization consists of replacing the integrands in Eq. (2) by their estimated values. In the following, we further consider an application of the EFD numerical scheme that was proposed for one- and two-dimensional steady-state and transient convection-diffusion problems in Cartesian coordinate system.

2.1 One-Dimensional Steady-State Case: EFD Formulation

The convection-diffusion process in a one-dimensional steady-state formulation can be described by the following conservation equation:

$$\frac{\partial(\rho u_x f)}{\partial x} = \frac{\partial}{\partial x}\left(\Gamma(x)\frac{\partial f}{\partial x}\right) + S(x), \tag{3}$$

where ρ is substance density, u_x is flow velocity in x direction, $\Gamma(x)$ and $S(x)$ are diffusion coefficient and source term both dependent on the x coordinate. Let us consider the control volume shown in Fig. 1. Integrating Eq. (3) over the given control volume gives the following

$$J(x_i + \Delta x/2) - J(x_i - \Delta x/2) = \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} S(x) dx , \tag{4}$$

where flux of f is defined as $J = \rho u_x f - \Gamma \frac{\partial f}{\partial x}$.

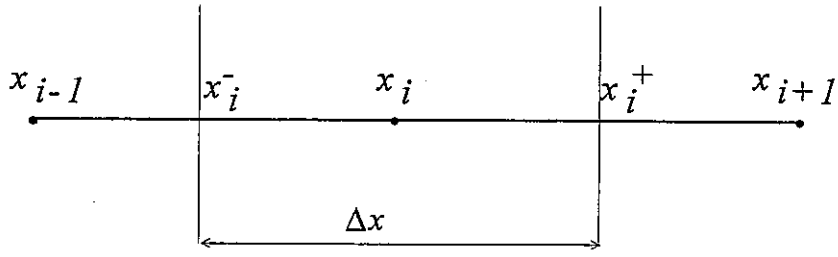

Figure 1. Typical one-dimensional control volume with uniform grid

Conventional upwind approximation of the interface fluxes in Eq. (4) for the left-hand side of control volume is as follows:

$$J^- = J(x_i^-) = \rho u(x_i^-) f_m - \Gamma(x_i^-)\frac{f_i - f_{i-1}}{\Delta x}, \tag{5}$$

where $x_i^{\pm} = x_i \pm \frac{\Delta x}{2}$, $m = \begin{cases} i-1, & u(x_i^-) \geq 0 \\ i, & u(x_i^-) < 0 \end{cases}$.

For positive u values, it leads to the following discretization of Eq. (3):

$$\rho u_i^+ f_i - \rho u_i^- f_{i-1} - \left(\Gamma_i^+ \frac{f_{i+1} - f_i}{\Delta x} - \Gamma_i^- \frac{f_i - f_{i-1}}{\Delta x} \right) = \bar{S} \Delta x \tag{6}$$

where superscript indices <+> and <–> denote the value at the corresponding $x_{i\pm}$ points. $\bar{S}$ is average value of S over the control volume.

It is well known that this upwind scheme discretization is most simple and it cannot predict an accurate solution for all range of Reynolds or Peclet numbers. The main problem is a "false" or numerical diffusion and a lot of improvements were proposed to suppress it. The exponential differencing (ED) scheme [1] takes into account local analytical solution of Eq. (3) derived for zero value of the source term between two adjacent control volumes. It results in the following expressions of interface fluxes:

$$J^- = \rho u_i^- \left(f_{i-1} - \frac{f_i - f_{i-1}}{\exp(Pe_i^-) - 1} \right), \qquad J^+ = \rho u_i^+ \left(f_i - \frac{f_{i+1} - f_i}{\exp(Pe_i^+) - 1} \right) \tag{7}$$

where the grid Peclet number is defined as $Pe_i^{\pm} = \frac{\rho u_i^{\pm} \Delta x}{\Gamma_i^{\pm}}$.

Recently, several other discretization schemes based on locally exact solutions have been proposed. Among them the LOAD scheme [3] takes into account the source term in Eq. (3) when an analytical solution is derived for the local control volume. It is assumed that this source term is constant over each control volume. The flux-spline discretization [4,9] is based on the assumption that within the control volume total flux in a given direction varies linearly along the coordinate. For example, the flux in the x-direction for the control volume around the grid point x_i (see Fig. 1) is given by

$$J(x) = \rho u_x f - \Gamma \frac{\partial f}{\partial x} = J^- + \frac{J^+ - J^-}{\Delta x} \left(x - x_i^- \right). \tag{8}$$

It means that the source term in Eq. (3) is considered as a constant within the control volume and, in fact, both, LOAD and flux-spline schemes, use the same basic assumption. The difference between them is way of their further treatment. Thus, the flux-spline scheme uses an iterative procedure to correct J^- and J^+ fluxes, while LOAD does not. Anyway, numerical results of both methods are the same in principle. Both schemes produce three-point linear equations for each node as following

$$a_i f_i + b_i f_{i-1} + c_i f_{i+1} + d_i = 0, \tag{9}$$

which allows to implement boundary conditions easily and to use standard matrix inversion algorithms like TDMA (Tree Diagonal Matrix Algorithm) or one-dimensional sweeping.

Another recently proposed scheme LENS [5] deals with the convection-diffusion equation with absorption term and takes into consideration parabolic source term distribution:

$$S(x) = a_0 + a_1 x + a_2 x^2. \tag{10}$$

This interpolation allows to obtain very accurate solution even on coarse grids however it should be mentioned that use of the interpolation scheme by Eq. (10) leads to four-point linear equations for each control volume that demands a specific solution procedure. Standard Gauss-Seidel iterations are also acceptable but efficiency of this algorithm is too poor to satisfy conditions of two- and three-dimensional problems commonly used in nuclear reactors analysis.

When deriving a locally analytical solution, all three discretizations in previous studies assume that diffusion coefficient Γ is a constant within a control volume: i.e., it does not depend on the coordinate. This assumption is incorrect in case of turbulent flows because turbulent viscosity and conductivity vary highly between adjacent grid nodes especially near the wall.

To overcome these limitations, EFD was proposed [7,8]. The development of the EFD scheme was presented at NURETH-8 conference [10] and later, the newest detailed description of EFD was published [11]. In this report, we discuss the improved implementation of EFD for calculation of turbulent flows with complex source terms and boundary conditions. At first, we will derive improved EFD formulation in a common case. In doing so, for one-dimensional steady-state problem we consider the following implementation of Eq. (3) to the local region within x_{i-1} and x_i shown in Fig. 1:

$$\frac{\partial(\rho u_x f)}{\partial x} = \frac{\partial}{\partial x}\left(\left(\Gamma^- + \left(x - x_i^-\right)\Gamma_i'\right)\frac{\partial f}{\partial x} \right) + S_i^- + \left(x - x_i^-\right)S_i', \tag{11}$$

where $\Gamma_i' = \dfrac{\Gamma_i - \Gamma_{i-1}}{\Delta x}, S_i' = \dfrac{S_i - S_{i-1}}{\Delta x}$.

Linear interpolation of the source term $S(x)$ and diffusion coefficient $\Gamma(x)$ is used here. With appropriate boundary conditions, namely values of f at i and i-1 points:

$$f(x_{i-1}) = f_{i-1}, \quad f(x_i) = f_i, \tag{12}$$

we obtain the problem to be solved for the local f distribution. What we finally need to derive is the value of the flux J^- at the interface between adjacent i and i-1 control volumes at the point x_{i-} as follows:

$$J^- = \left(\rho u_x f - \Gamma(x)\frac{\partial f}{\partial x} \right)_{x = x_i^-}. \tag{13}$$

Integration of Eq. (11) gives the following:

$$\rho u_x f = \left(\Gamma^- + \left(x - x_i^-\right)\Gamma_i'\right)\frac{\partial f}{\partial x} + \left(x - x_i^-\right)S_i^- + \frac{\left(x - x_i^-\right)^2}{2}S_i' + J^-. \tag{14}$$

Common solution of Eq. (14) can be written as

$$f(x) = C\left(\Gamma^- + \left(x - x_i^-\right)\Gamma_i'\right)^{Pe_i^- \frac{\Gamma^-}{\Gamma_i - \Gamma_{i-1}}} + \frac{J^- + \Gamma^- a_1}{\rho u_x^-} + a_1\left(x - x_i^-\right) + a_2\left(x - x_i^-\right)^2 \tag{15}$$

where $a_1 = \dfrac{S_i^- + 2a_2\Gamma^-}{\rho u_x^- - \Gamma_i'}$ and $a_2 = \dfrac{S_i - S_{i-1}}{2\left(Pe_i^- - 2\dfrac{\Gamma_i - \Gamma_{i-1}}{\Gamma^-} \right)}$.

Constants of integration C and J^- should be found from boundary conditions (12). For the left-side interface flux, it gives

$$J^- = \rho u_i^- f_{i-1} - \tilde{K}\Gamma^- \frac{f_i - f_{i-1}}{\Delta x} + \tilde{S}, \tag{16}$$

where $\tilde{K} = Pe_i^- \left(\left(\frac{\Gamma_i}{\Gamma_{i-1}} \right)^{Pe_i^- \frac{\Gamma^-}{\Gamma_i - \Gamma_{i-1}}} - 1 \right)^{-1}$,

$$\tilde{S} = D\frac{S_i + S_{i-1}}{2}\Delta x + (S_i - S_{i-1})\Delta x \frac{D - Pe_i^-/8}{Pe_i^- - 2\frac{\Gamma_i - \Gamma_{i-1}}{\Gamma^-}}$$

and $$D = \frac{\frac{Pe_i^-}{2} - 1 + \tilde{K}}{Pe_i^- - \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma^-}}.$$

It is easy to see that Eq. (16) differs from simple upwind discretization given by Eq. (5) in a corrected diffusion coefficient $\tilde{K}$ and a new source term $\tilde{S}$ depending on the source distribution near the grid node. While $\tilde{K}$ depends only on the mesh Peclet and diffusion coefficient distribution, $\tilde{S}$ depends also on spatial source term distribution. It should be noted here that, in some special cases, including of standard constant diffusion coefficient Γ distribution, Eq. (16) gives "zero division" numerical errors when applied directly in a computer code and thus the limit should be taken.

Final EFD formulation consists of substitution of Eq. (16), derived for both left- and right sides of control volume, to the balance Eq. (4). Then, the integral in the right-hand-side of Eq. (4) should be approximated also. Commonly we use a three-point parabolic interpolation that yields the following:

$$\int_{x_i-\Delta x/2}^{x_i+\Delta x/2} S(x)dx = S_i\Delta x + \frac{S_{i+1} - 2S_i + S_{i-1}}{24}\Delta x. \tag{17}$$

As a result of substitution of Eq. (17), Eq. (16) and corresponding equation written for the right-side of the control volume into the Eq. (4), we obtain a standard three-point linear equation for the given grid point as following:

$$a_i f_{i-1} + b_i f_i + c_i f_{i+1} + d_i = 0, \tag{18}$$

where d_i could be written as $d_i = A_i S_{i-1} + B_i S_i + C_i S_{i+1}$.

Equation (18), when written for each internal grid node with appropriate boundary conditions, gives a system of linear equation to be solved.

3. ONE-DIMENSIONAL SAMPLE PROBLEMS

3.1 Linear Distribution of Source Term

To examine the accuracy of the scheme described so far when dealing with nonuniform source term, use is made of a sample problem in one-dimension defined by Eq. (3), where

$$S(x)=\begin{cases} S_0\left(\frac{1}{2}-x\right), & 0\le x\le\frac{3}{4} \\ S_0(x-1), & \frac{3}{4}\le x\le 1 \\ 0, & 1\le x\le\infty \end{cases}, \tag{19}$$

with the boundary conditions $T(0)=T(2)=0$. Similar problem was originally considered by Leonard [2] and Prakash [6] to examine efficiency of QUICK and LOAD schemes. Figure 2 shows a comparison of numerical results of the relevant schemes with an exact analytical solution.

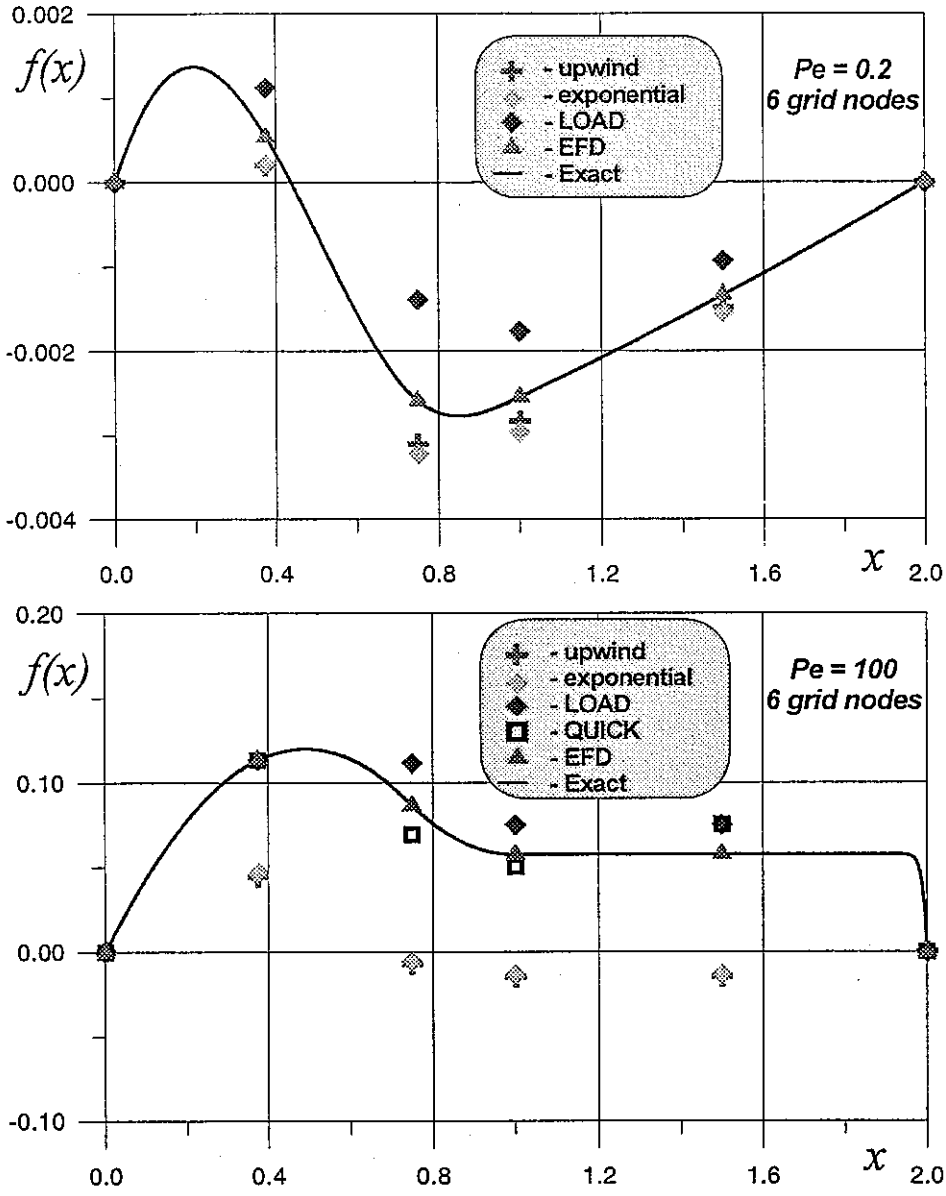

Figure 2. Comparison of numerical results calculated by upwind, exponential, LOAD, QUICK and EFD for one-dimensional steady-state problem using 6-nodes mesh; Pe = 0.2 (a); Pe = 100 (b)

With all methods numerical solutions can be calculated for all Peclet numbers from 0.2 to 1000 except for the QUICK scheme. Our realization of QUICK was found to become unstable for the small mesh Peclet numbers. All schemes also predict solutions close to the exact solution with large grid point numbers. For example, when calculated for $Pe=1000$, to achieve the same accuracy of $\varepsilon = 10^{-3}$, where the accuracy index ε is defined by

$$\varepsilon = \frac{1}{N_{points}} \sum_{i=1}^{N_{points}} \left| f(x_i)^{numerical} - f(x_i)^{exact} \right|, \qquad (20)$$

upwind and exponential schemes require about 800 grid points; LOAD and QUICK ~30 grid points; and EFD uses 6 points only. This result does not surprise because EFD should produce an exact

solution if at least one grid node is placed to each point linking the adjacent region of the source term defined by Eq. (19) for this kind of the source term distribution, namely linear functions. For the source term defined by Eq. (19), corresponding points are $x = 0,\ 0.75,\ 1.0$ and 2.0.

3.2 Adding Absorption Term

The sample problem considered in above section is easy to be solved by the EFD because the source term distribution given by Eq. (19) is linear. It correlates exactly with the main assumption of the EFD scheme that the source term varies linearly between adjacent grid nodes. To highlight advantages of EFD, while running under complex conditions, another problem has been chosen to solve one-dimensional steady-state convection-diffusion equation with more complex source term and with additional absorption term given by the following

$$Pe\frac{\partial f}{\partial x} + Pf = \frac{\partial^2 f}{\partial x^2} + Q(x), \tag{21}$$

with boundary conditions $\{f(0)=1,\ f(1)=0\}$. This problem was originally considered by Sakai [5] to check the LENS scheme especially developed to manage absorption and source terms. The finite-difference discretization of Eq. (21) by the EFD scheme for the given control volume i is written in the following

$$J_i^+ - J_i^- + Pf_i\Delta x = \int_{\Delta x} Q(x)dx, \tag{22}$$

where J_i^+ and J_i^- are defined by Eq. (16) and the right-hand-side integral of the source term Q is approximated by Eq. (17). It should be mentioned that when we calculate the interface fluxes given by Eq. (16) compound source term S is used as follows

$$S_i = Pf_i^{\nu} + Q_i. \tag{23}$$

Whereas values of S_i used at three points $i,\ i+1,\ i-1$ depend on function f_i^{ν} spatial distribution, we should use an iterative procedure to correct source term S at these grid nodes. Thereby, the superscript ν in Eq. (23) refers to the previous value of f. It has been shown that two or three iterations are enough to reach the accuracy of 0.1% for problem under the test.

Figure 3 shows the distribution of f in the x-direction in case of large absorption term $P=1000$ and source term given by $Q(x)=(x-0.5)^2$ for Peclet number $Pe=500$. This sample was considered by Sakai [5] (see p.65, Fig. 8) and results calculated by the QUICK and LENS schemes have been taken from that figure. The 16-nodes grid was used by Sakai [5] but the EFD has been tested on 16- and 6-nodes meshes. A comparison shows that EFD has already reached good agreement with the exact solution using 6 grid points and numerical solution is practically coincides with analytical one for 16-points grid. In this particular case, solution calculated with EFD is better than that predicted by LENS.

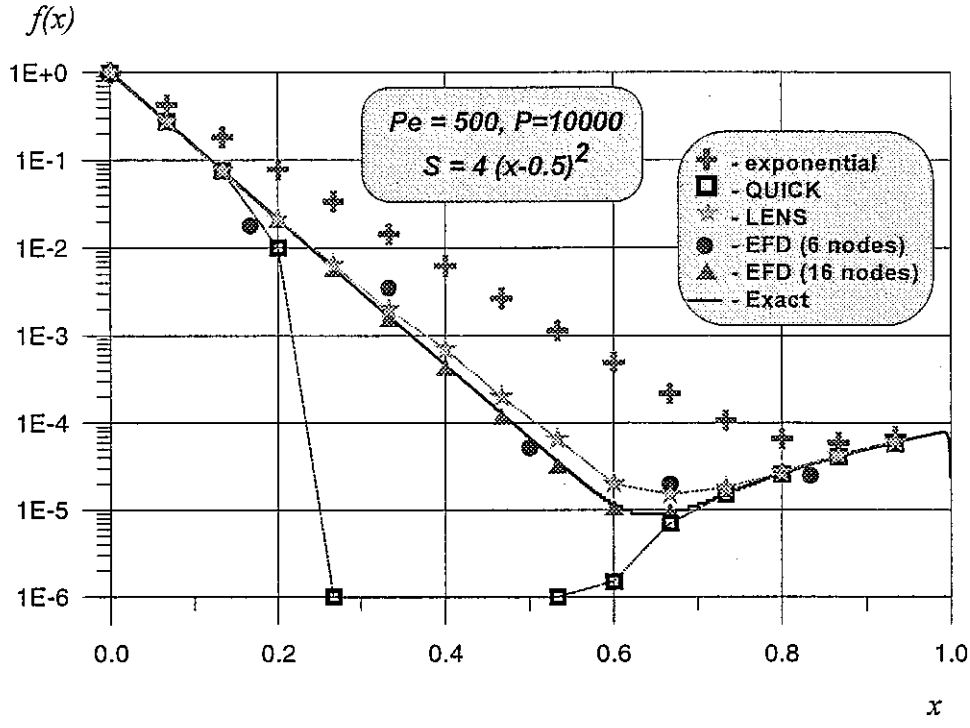

Figure 3. Comparison of numerical solutions of Eq. (21) with exact analytical solution; Results calculated with EFD are shown on 16- and 6-nodes grids; others for 16-nodes grid

We would like to note here that the LENS scheme was especially developed to handle equations with large absorption [5]. The application of EFD is more universal, namely, it works with all types of convection-diffusion transport equations. While LENS takes into consideration locally exact analytical solution of Eq. (21) with the absorption term separated from others, EFD simply treats that term as an extra source. Nevertheless, EFD predicts a more accurate solution even in this special case.

3.3 Transient Problems: Initial Step Propagation

To apply the EFD scheme to transient transport problems, we consider the transient term as an extra source and follow the procedure similar to that used to describe the absorption term behavior in the steady-state problem considered above. Namely, in order to obtain a discretization of the following one-dimensional transient convection-diffusion equation

$$\frac{\partial(\rho f)}{\partial \tau}+\frac{\partial(\rho u_x f)}{\partial x}=\frac{\partial}{\partial x}\left(\Gamma(x)\frac{\partial f}{\partial x}\right)+Q(x), \tag{24}$$

we integrate it over the control volume (shown in Figure 1.) and use Eq. (16) for the interface heat fluxes on the left and the right side. Using a full-implicit form of a transient term, we obtained the following finite-difference discretization of Eq. (24):

$$\rho\left(f_i^k-f^{k-1}\right)\frac{\Delta x}{\Delta \tau}+J_i^+-J_i^-=\int_{\Delta x}Q(x)dx, \tag{25}$$

where $\Delta\tau$ is the time increment; and indices k and k-1 refer to values on current and previous time steps, τ_k and τ_{k-1} respectively. We also use Eq. (17) to represent the integral at the right-hand-side

of Eqs. (25) and (16) to derive J_i^+ and J_i^- interface fluxes. In doing so, we define the source term S_i in Eq. (25) as

$$S_i = Q_i - \rho \frac{f_i^{k^v} - f_i^{k-1}}{\Delta \tau}. \tag{26}$$

This value should be corrected by iterations and superscript index V refers to the latest iteration step. It has been found that two or three iterations are enough to achieve a good convergence. Of course, it depends on a problem to be solved, but for all problems considered this algorithm has shown a good convergence.

To check EFD and compare the accuracy, we take a sample problem where an initial sudden step of transported quantity is propagated by the flow in one dimension. It is described by the governing equation (24) where the source term $Q(x)=0$ with following initial and boundary conditions

$$\begin{aligned} f(x,0) &= 0, \\ f(0,\tau) &= 1, \\ f(L,\tau) &= 0. \end{aligned} \tag{27}$$

It is well known that for this kind of convection-diffusion problems, when one analyzes using ordinary methods such as upwind or the exponential schemes, it is difficult to obtain the accurate numerical solution due to the numerical diffusion. Figure 4 shows a calculated distribution of $f(x, \tau)$ in x-direction for $\tau/\rho = 0.005$ and $\tau/\rho = 0.00005$, and for the Peclet numbers 100 and 10000 respectively. Here, Peclet number Pe is defined as $\frac{\rho u L}{\Gamma}$.

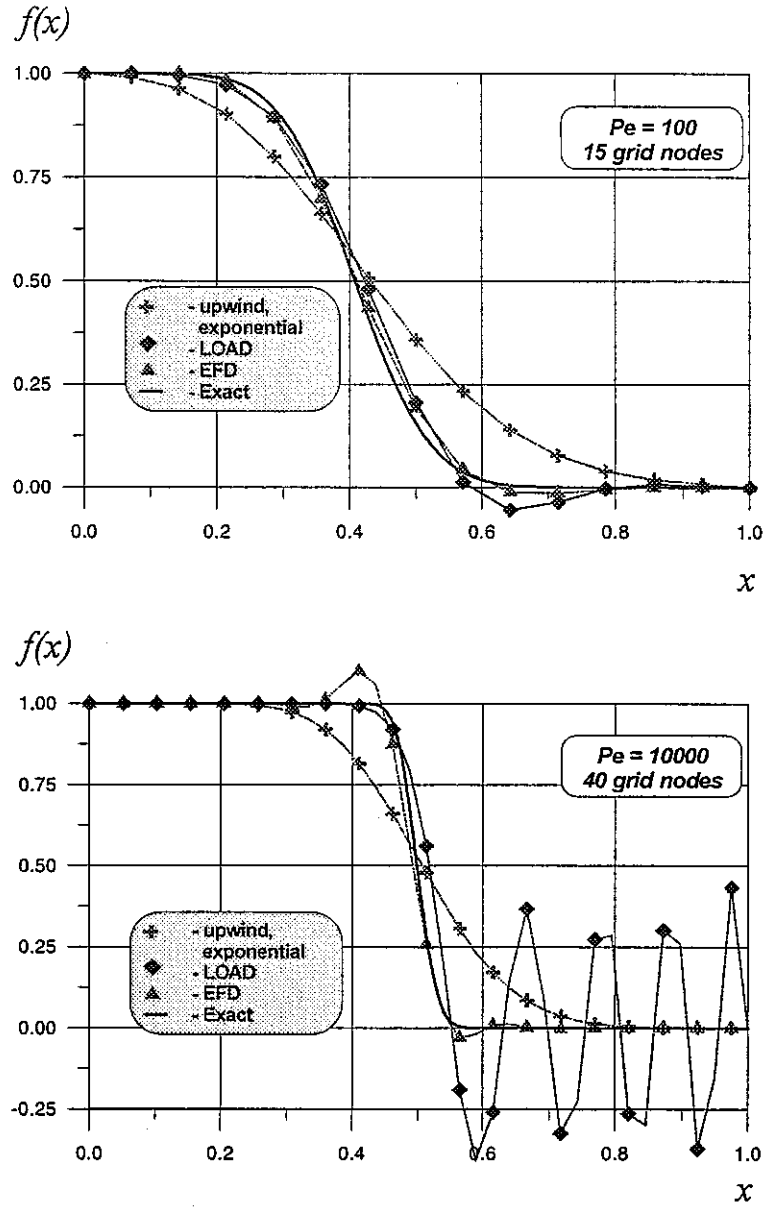

Figure 4. Propagation of the sudden initial step in one-dimension; Standard EFD and LOAD implementation

As shown in Fig. 4 the upwind and the exponential schemes generate a large numerical diffusion while EFD and LOAD do not. However, both EFD and LOAD tend to produce unphysical oscillations for large Peclet numbers. When calculating by the LOAD scheme, numerical solutions start to oscillate visibly for $Pe=100$, and, for $Pe=10,000$, cannot be trusted.

3.4 Transient Problems: Eliminating of Unphysical Oscillations

EFD also generates visible oscillations for $Pe=10,000$. However, it shows good agreement with the exact solution and produces the smallest average numerical error among the schemes tested, whereas error was defined by Eq. (20). It should be noted that both EFD and LOAD remain to be stable and converge in a few iteration steps. We have found the reason for those oscillations as following. When deriving an approximation of the interface fluxes according

to Eq. (16), we have assumed first, that the source term distribution is defined by Eq. (26) and, second, it should be calculated by iterations. Neither EFD, applying the linear source term distribution, nor LOAD, assuming the source term as a constant within the control volume, cannot describe correctly a distribution of the time-derivative term in Eq. (24) on a coarse mesh when a function of a transported quantity behaves as a sudden sharp step. It leads to a strong domination of the correction term $\tilde{S}$ in Eq. (16) over the others. This term depends on the function distribution according to the Eq. (26) and should be corrected by iterations. It is easy to expect that, in some cases, this algorithm could involve incorrect negative or positive false sources into the finite-difference discretization of the governing equation (24). In generally, when a linear interpolation cannot describe the source term distribution correctly, EFD may generate an unphysical solution. Nevertheless, this solution is close to exact one and the numerical error is the smallest among all others finite-difference schemes tested in this report.

However, it is important to prevent any unphysical behavior of the transported quantity and one could apply a number of appropriate techniques to eliminate or, at least, to reduce these effects. For example, it is a common practice to use the upwind scheme for grid points where oscillations occur or other symptoms indicate an unphysical numerical solution. It should be noted, when real problems are calculated, it is difficult to separate numerical oscillations from physical and meaningful fluctuations of the transported quantity. Thereby, a different algorithm has been applied in this report. If a mesh and a sharp shape of the quantity do not allow us to interpolate the source term distribution correctly, we could apply a restriction that suppresses an undesirable domination of $\tilde{S}$. In short, the main point is to find the grid nodes where the correction term $\tilde{S}$ dominates on the main term in formula (16), which is defined as $\rho u_i^- f_{i-1} - \tilde{K}\Gamma^- \frac{f_i - f_{i-1}}{\Delta x}$ and set $\tilde{S}$ equal to this value. It was found that this simple algorithm, being applied to a problem of initial step propagation, eliminates the oscillations for both LOAD and EFD scheme. Furthermore, an average numerical error generated by EFD does not grow much. Results calculated by adjusted EFD and LOAD scheme are shown in Fig. 5.

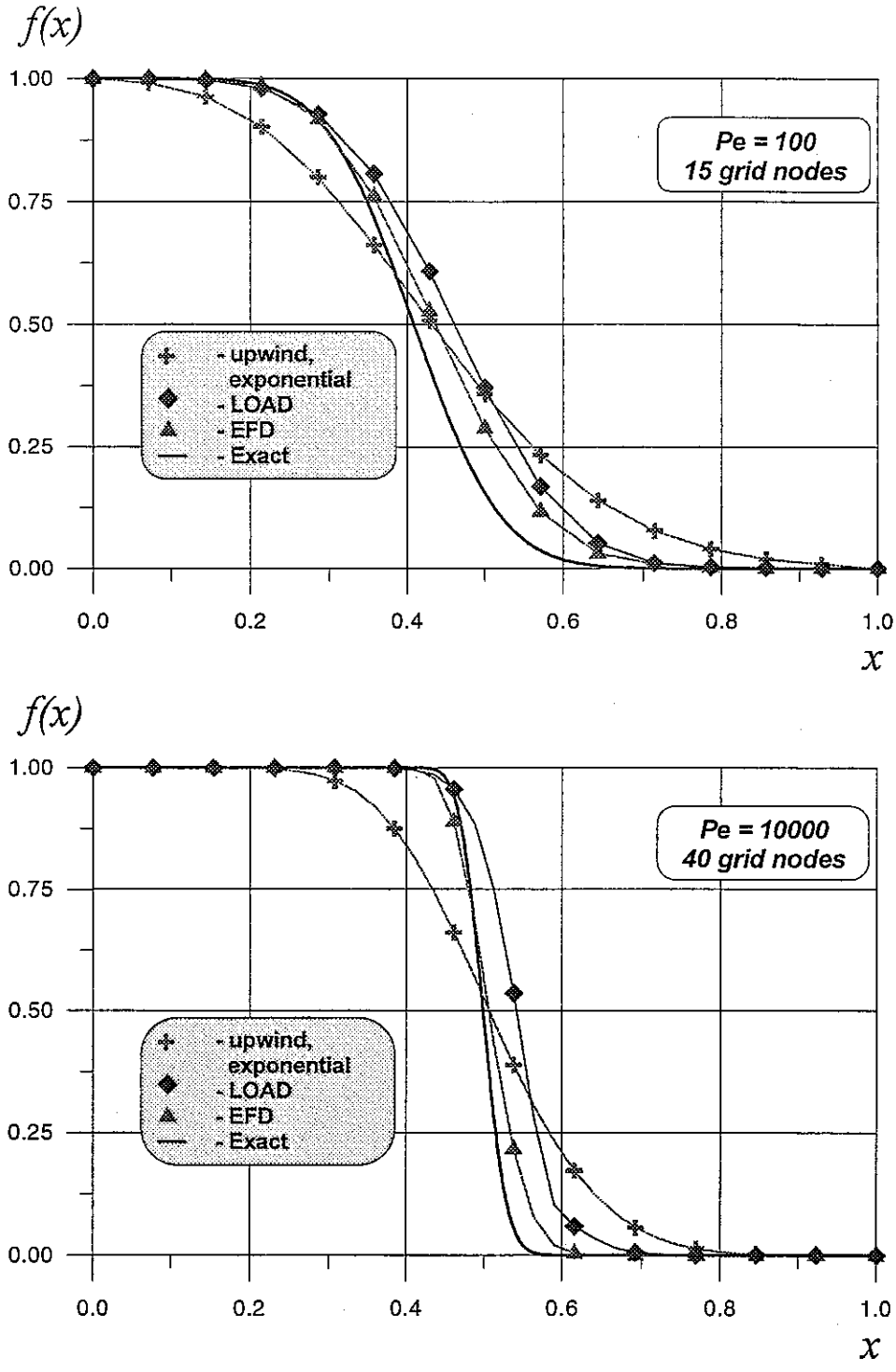

Figure 5. Propagation of the sudden initial step in one-dimension; Extra source terms in EFD and LOAD are limited

Neither EFD nor LOAD methods, corrected by simple algorithm above, generate unphysical oscillations, although an average numerical error of the EFD scheme has slowly increased for calculation with $Pe = 100$. Nevertheless, EFD produces the best numerical solution of all schemes tested in a range of all Peclet numbers considered. In our opinion, this sample

demonstrares the ability of the EFD scheme to give an accurate numerical solution for problems which may be considered as a "hard" to check a numerical technique.

It should be mentioned that we did not compare CPU time for one-dimensional samples calculated by different methods. First, it is difficult to compare methods which use the different algorithms to invert the matrices. For the upwind, the exponential, the LOAD and the EFD schemes we used direct methods while for QUICK a simple iterative procedure was implemented. Data for comparison with LENS were captured directly from the paper of Sakai [5]. Second, and more important reason is the following. In fact, it is not very valuable to compare the computational efforts for one-dimensional problems because the real problems are two- and three-dimensional. In these cases, extra iterations increasing CPU time should be applied for locally analytical schemes such as LOAD, LENS and EFD.

4. TWO-DIMENSIONAL PROBLEMS

4.1 Application of the EFD Scheme in Two- and Three-Dimensions

The one-dimensional EFD scheme can be easily extended to two- and three-dimensional problems. In doing so, we consider a change of flux in one direction as a source term in other directions. This is very similar to the procedure we applied on transient term manipulation. An interface flux expression (16) resulting from the EFD scheme formulation contains a specific term $\tilde{S}$ that appears due to the assumption of a linear variation of a total source term in one-dimension within a control volume. The additional source term involves differences of fluxes at adjacent faces of the control volume. That the fluxes in a given directions at two faces of the control volume (J^+ and J^-) are different indicate the presence of physical sources of the transported quantity and/or multidimensionality. Expression such as Eq. (16) can also be derived for interface fluxes in the other coordinate directions. For example, the discretization of Eq. (2) over the control volume surrounding grid point i,j in two-dimension Cartesian system could be written as

$$\rho\left(f_{i,j}^{k}-f_{i,j}^{k-1}\right)\frac{\Delta x\Delta y}{\Delta\tau}+\left(J_{i,j}^{x+}-J_{i,j}^{x-}\right)\Delta y+\left(J_{i,j}^{y+}-J_{i,j}^{y-}\right)\Delta x=\iint_{S}Q(x,y)dxdy, \tag{28}$$

where $J_{i,j}^{x+}, J_{i,j}^{x-}, J_{i,j}^{y+}, J_{i,j}^{y-}$ are the total fluxes of transported quantity f in x- and y- direction at corresponding surfaces of the control volume as shown in Fig. 6. These fluxes are defined as

$$\begin{aligned}
J_{i,j}^{x+} &= \rho u_x f\left(x_i+\tfrac{\Delta x}{2}, y_j\right)-\Gamma\frac{\partial f}{\partial x}\Big|_{x_i+\frac{\Delta x}{2},y_j}, \\
J_{i,j}^{x-} &= \rho u_x f\left(x_i-\tfrac{\Delta x}{2}, y_j\right)-\Gamma\frac{\partial f}{\partial x}\Big|_{x_i-\frac{\Delta x}{2},y_j}, \\
J_{i,j}^{y+} &= \rho u_y f\left(x_i, y_j+\tfrac{\Delta y}{2}\right)-\Gamma\frac{\partial f}{\partial y}\Big|_{x_i,y_j+\frac{\Delta y}{2}}, \\
J_{i,j}^{y-} &= \rho u_y f\left(x_i, y_j-\tfrac{\Delta y}{2}\right)-\Gamma\frac{\partial f}{\partial y}\Big|_{x_i,y_j-\frac{\Delta y}{2}}.
\end{aligned} \tag{29}$$

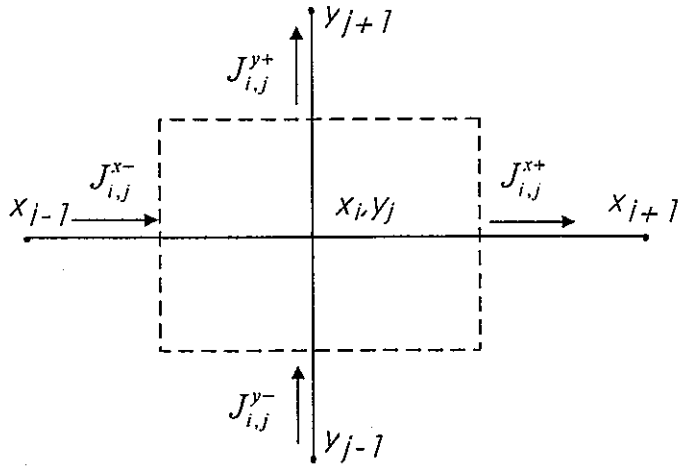

Figure 6. Typical two-dimensional control volume in the Cartesian coordinate system

Since Eq. (16) defines the finite-difference approximation of the interface fluxes, it can be written for all surfaces of the control volume. Substituting these expressions into Eq. (29), the following full implicit discretization equation is derived:

$$a_{ij} f^k_{i,j} + b_{ij} f^{k-1}_{i,j} + a^{x-}_{ij} f^k_{i-1,j} + a^{x+}_{ij} f^k_{i+1,j} + a^{y-}_{ij} f^k_{i,j-1} + a^{y+}_{ij} f^k_{i,j+1} + c_{ij} = 0, \qquad (30)$$

where k and k-1 refer to τ and τ-$\Delta\tau$ time steps; a's represent the ordinary coefficients, which can be easily derived from Eq. (16); b_{ij} represents the time-dependence coefficient; and c_{ij} is a sum of a physical source term and additional contribution from the EFD formulation, given by

$$c_{ij} = Q_{ij}\Delta x\Delta y + \left(\tilde{S}^{x+}_{ij} - \tilde{S}^{x-}_{ij}\right)\Delta y + \left(\tilde{S}^{y+}_{ij} - \tilde{S}^{y-}_{ij}\right)\Delta x \qquad (31)$$

Whereas every $\tilde{S}$ term depends on the spatial interface flux distribution, which also depends on the transported quantity distribution, an iteration algorithm should be applied to resolve the system of equations (30) written for each grid points i,j. We start with the initial values of fluxes and solve for the transported quantity distribution $f(x,y)$ at all the grid nodes. Then, the new values of interface fluxes are computed. Thus, the system of linear equation should be solved on every iteration. For the two-dimensional case we come up with a 5-node finite-difference scheme. This system can be solved by conventional methods such as the two-dimensional line-by-line sweeping based on the Three-Diagonal Matrix Algorithm (TDMA).

A computer program that implements a common discretization of three-points locally analytical schemes has been developed. This algorithm allows using those methods only which produce three-point discretization in one dimension to make it possible to apply standard line-by-line TDMA. This limitation does not allow to incorporate those schemes which use four and more grid nodes in one dimension for discretization. Therefore, we implemented the upwind, the exponential, the LOAD and the EFD schemes and did not implement QUICK and LENS because TDMA was used as a basic matrix procedure in line-by-line sweeping.

The computer program is written in C++ in the portable form with user interface running on PCs under the Windows NT operation system. An Object-Oriented Programming technique [12] is used to calculate a general convection diffusion problem. First, we define a single base class which describes all the properties of standard finite-difference methods including grid generation, matrix operations, input/output, etc.. Second, we derive from that base class a few child ones which

implement corresponding formula for interface flux discretization according to the numerical schemes to be implemented. Thereby, the child classes inherit all the properties of the base class with possibility to calculate whole the problem but differ in a single function (method) only which defines the formula for discretization coefficients. For example, Eq. (5) is used for the upwind scheme; Eq. (7) for exponential and Eq. (16) for EFD. It allows correct comparison of CPU time and other parameters of the several numerical schemes in the same code.

4.2 Fully-Developed Turbulent Flow in Plane Channel: Heat Transfer

As a sample, we consider a heat transfer problem in a plane channel of turbulent flow with the corresponding boundary conditions for a two-dimensional area.

In order to check an efficiency of EFD for two-dimensional convection-diffusion problems and to compare numerical results with an exact analytical solution, we use the Cartesian coordinate system (x_1,x_2) rotated at an angle of 45° relative to the regular (x,y) system as shown in Fig. 7.

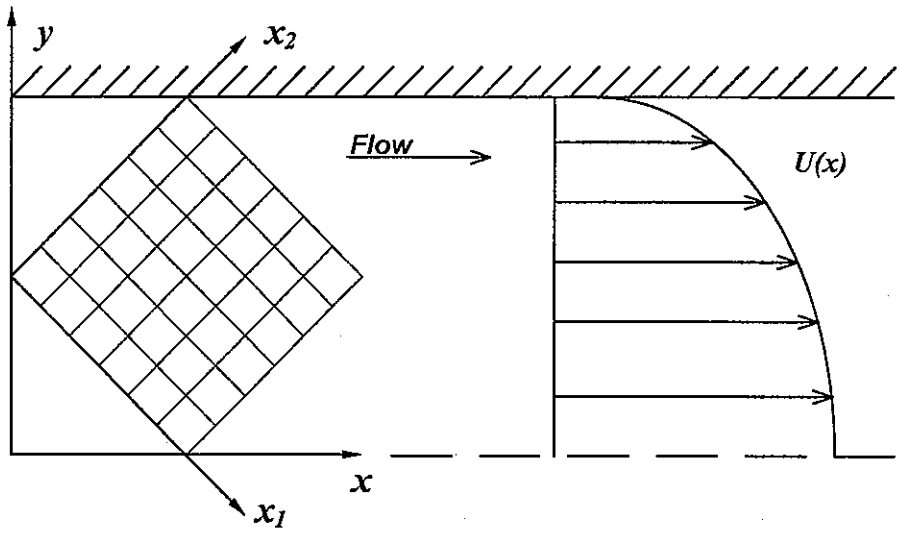

Figure 7. Calculation of the turbulent flow in a plane channel in the Cartesian coordinate system rotated at an angle of 45°.

A steady-state fully-developed energy conservation problem has been chosen as a sample because it is easy to derive an analytical solution when the boundary condition is the fixed heat flux on the wall. The temperature distribution for this problem $T(x,y)$ may be written as $t(y) + const \times x$. In fact, it is an one-dimensional profile varying linearly in x-direction. The temperature profile $t(y)$ can be easily found for fully-developed flow and heat transfer.

The energy conservation equation in the transformed coordinate system can be written as

$$\frac{\partial(\rho c_p w_1 T)}{\partial x_1}+\frac{\partial(\rho c_p w_2 T)}{\partial x_2}=\frac{\partial}{\partial x_1}\left(\left(\lambda+\lambda^T\right)\frac{\partial T}{\partial x_1}\right)+\frac{\partial}{\partial x_2}\left(\left(\lambda+\lambda^T\right)\frac{\partial T}{\partial x_2}\right), \qquad (32)$$

where $w_1(x_1,x_2)=\frac{\sqrt{2}}{2}u(y)$; $w_2(x_1,x_2)=\frac{\sqrt{2}}{2}u(y)$; $u(y)$ is the velocity distribution in turbulent flow; and $\lambda^T(y)$ is the turbulent thermal conductivity that, as we assume for simplicity, corresponds directly to the turbulent viscosity $\nu^T(y)$. This assumption states that turbulent Prandtl number $Pr^T=\rho c_p \nu^T/\lambda^T=1$. The distribution of the turbulent viscosity can be derived from

the axial velocity distribution. In this report, a simple three-layer universal velocity profile is used as follows:

$$\begin{aligned} u/u_* &= y^+, && y^+ \le 5 \\ u/u_* &= 5\ln y^+ - 3.05, && 5 \le y^+ \le 30 \\ u/u_* &= 2.5\ln y^+ + 5.5, && y^+ \ge 30 \end{aligned} \quad , \tag{33}$$

where $u_* = \sqrt{\zeta/8}$ is the dynamic velocity; ζ is the friction factor; $y^+ = yu_*/v$ is the non-dimensional distance to the wall. Under these conditions, the governing equation (32), when written in the original one-dimensional form, can be easily integrated numerically with sufficiently many points of integration. Due to the one-dimensional integration, we can use as many points as necessary to ensure the numerical solution is very precise and close to the analytical one. In theb following, we will refer to it as an "exact" solution.

Boundary conditions for the two-dimensional sample problem are given by the exact one-dimensional solution above. Governing equation (32) is discretitazed by the upwind, exponential, QUICK, LOAD and EFD schemes. The corresponding systems of linear equations were solved using the line-by-line TDMA. The same computer code was used for all the methods tested. Iterations are stopped when the maximum relative difference between current value and the one from latest iteration reaches the predefined error value:

$$\varepsilon_{stop} > \underset{i=1,j=1}{\overset{N_{points}}{MAX}} \frac{\left| T_{i,j} - T(x_1^i, x_2^j)^{last} \right|}{\left| T(x_1^i, x_2^j) \right|}. \tag{34}$$

In most ordinary cases, when the total number of grid points is below 70x70, ε_{stop} can be chosen as 1.0E-3. However, for bigger numbers of grid points, ε_{stop} should be smaller to assure that the slow convergence rate does not affect the final solution. This is especially important for precise schemes, such as QUICK, LOAD and EFD. It has been found, that for huge numbers of grid points (bigger than 70x70), when these schemes reach relative accuracy about 1.0E-4, additional increasing of grid nodes does not improve the accuracy. In these cases, the value of ε_{stop} should be decreased and decreased significantly. Whereas the calculations included grids with sizes up to 150x150, stop value ε_{stop} was given as low as 1.0E-8.

In addition, as the QUICK method demands an extra grid point upstream, a special treatment of the upstream boundary should be used. The boundary value itself is given directly by the exact solution. For simplicity, the second node value could be also given by the exact solution at the corresponding node. However, in this case, the comparison of methods considered would not be correct because the exact information about the second node values is not used by other methods. In fact, it has been found, that simplification decreases (improves) an average error of the QUICK scheme by 20-50%. To make it sure the comparison is correct, more complicated but proper upstream boundary condition proposed in the original work of Leonard [2], is used. For the QUICK computation, the gradient is also needed at the upstream end of the computational region. This upstream boundary gradient is chosen to be consistent with quadratic interpolation between the given boundary value T_0 and the first two interior node values T_1 and T_2 as follows:

$$\left.\frac{\partial T}{\partial x_i}\right|_{bound} = \frac{4T_1 - T_2 - 3T_0}{2\Delta x_i}, \tag{35}$$

where Δx_i is the step value of the uniform grid in the x_i direction.

In addition, the line-by-line TDMA algorithm is slightly changed for QUICK scheme. The contribution of the extra point is transferred to the source term along with terms from other directions. It has been found that it does not increase the total number of iterations to reach convergence. Total CPU time is slightly increased but not significantly.

Figure 8 shows a comparison of calculated results in terms of relative errors compared with the exact analytical solution, where an average relative numerical error ε is defined as

$$\varepsilon = \frac{1}{N_{points}} \sum_{i=1,j=1}^{N_{points}} \frac{\left| T_{i,j}^{\ numerical} - T(x_1^i, x_2^j)^{\ exact} \right|}{\left| T(x_1^i, x_2^j)^{\ exact} \right|}. \tag{36}$$

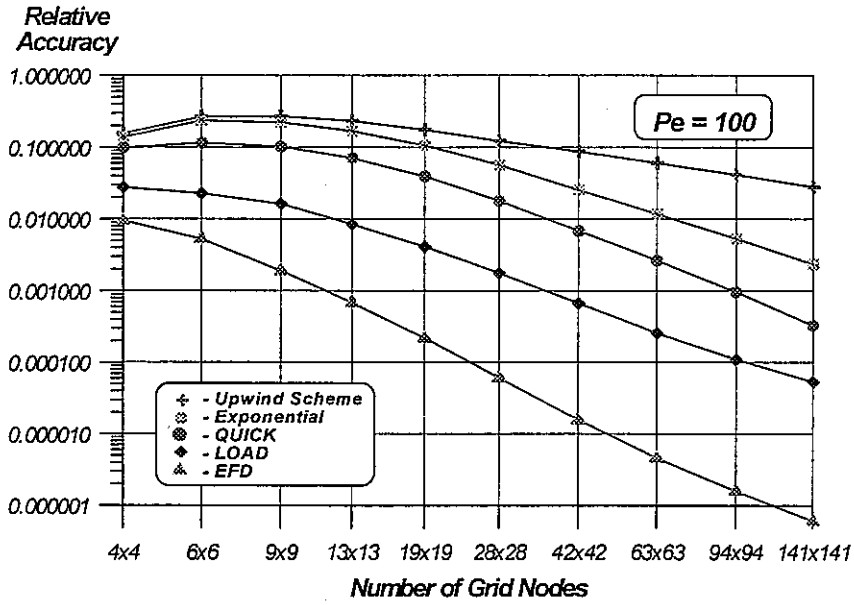

Figure 8. Two-dimensional test problem: variation of average relative error with number of grid points in x- and y-directions

The results shown in Fig. 8 were calculated for the fully-developed turbulent flow of liquid metal in a plane channel for Peclet number $Pe=100$.

The upwind and exponential schemes perform very poor. The false numerical diffusion seems to be an obvious reason. QUICK is significantly (about an order of magnitude) better but still not comparable with EFD. EFD demonstrates much better results than LOAD and QUICK for the turbulent flows. This is considered to be due to the fact that the EFD scheme takes into account the distribution of both the source term and the diffusion coefficient within every control volume. LOAD treats a source term as a constant within the control volume but assumes the diffusion coefficient is also a constant between two adjacent grid nodes.

It should be noted again that in this problem the "source term" does not represent any physical source term but the sum of partial derivatives from another direction.

In fact, the difference (in terms of numerical error) between constant and linear source term distributions is not very big. It can be compared with numerical integration using either rectangular formula (zero-order approximation) or trapezoid rule (first-order approximation). While the later is supposed to be more accurate, the difference is often insignificant. Even more, sometimes rectangular formula integrates even more accurate. Only the second-order Simpson method

improves numerical integration significantly. It is interesting to note here, that the Simpson method can be considered similar to QUICK.

However, the winning benefit of EFD is the linear interpolation of diffusion coefficient inside the control volume (along with considering the source term distribution as well). That is why EFD performs better for turbulent flows. Regular methods such as upwind and exponential schemes demand very fine mesh system close to the walls because both the transported quantity (velocity and/or temperature) and turbulent diffusivity profiles vary sharply near the wall region.

Figure 9 shows a comparison of the CPU time vs. numerical error. It is easy to see that with EFD the numerical error is a few orders of magnitude smaller than that of other schemes considered. Alternatively, in order to attain the same level of specified numerical error, EFD requires less CPU time by several orders of magnitude.

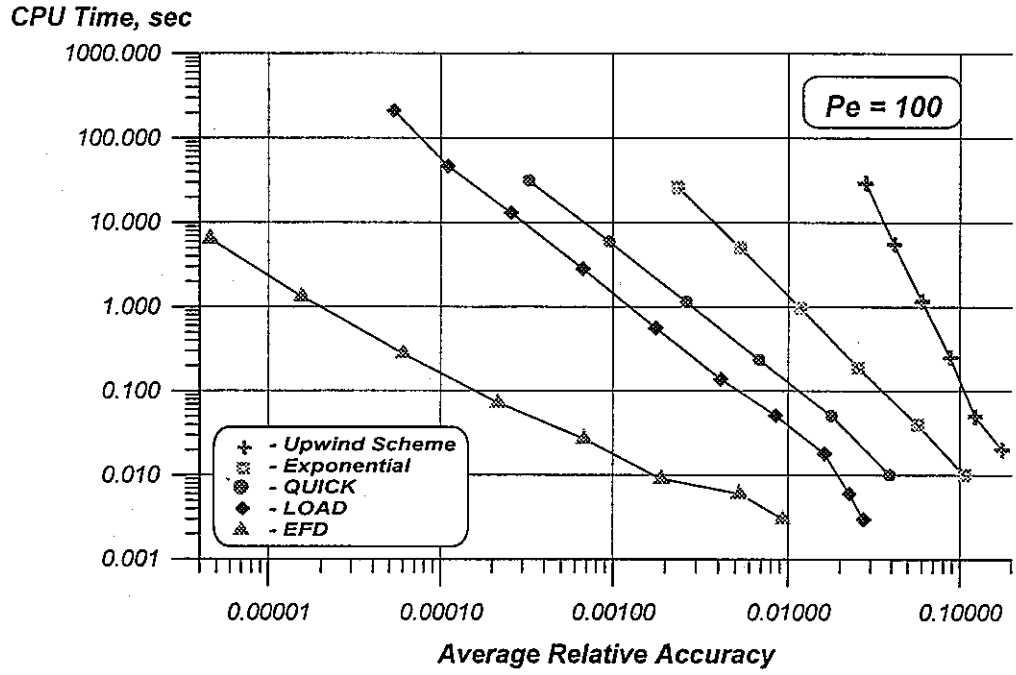

Figure 9. CPU time vs. numerical error (Pe=100)

Numerical values of two-dimensional sample calculations are summarized in Table 1. One can see that LOAD and EFD schemes demand about twice iterations compared with the regular upwind and exponential, as well as the QUICK schemes. In addition, in the case of EFD, the CPU time per grid node is three times longer. LOAD takes even five times longer (per node) due to many exponential calculations in this scheme. As a result, when locally-analytical methods are compared with regular ones, the total CPU time is longer as much as five times in the case of EFD and about eight times for the LOAD scheme.

Nevertheless, those additional computational efforts are well compensated by higher, a few orders of magnitude, accuracy. In terms of CPU time per given numerical error, the EFD scheme shows the best performance for this sample problem.

These results are not very surprising because it is observed many times that the "false" numerical diffusion plays very significant role especially in these mesh nodes where grid lines do not coincide with the flow direction. However, as it was mentioned above, in these cases, EFD can predict an accurate numerical solution.

Table 1 Accuracy ε, number of iterations IT, and CPU time, sec.
(Pentium II-400Mhz, Borland C++ v.4.0 compiler)

grid		Upwind	Exp.	QUICK	LOAD	EFD
6x6	ε	**0.2658**	**0.2339**	**0.1149**	**0.0226**	**0.0053**
	IT	6	3	18	17	18
	CPU	<0.001	<0.001	<0.001	0.006	0.006
9x9	ε	**0.2702**	**0.2202**	**0.1014**	**0.0162**	**0.0019**
	IT	9	4	19	20	20
	CPU	<0.001	<0.001	<0.001	0.018	0.009
28x28	ε	**0.1217**	**0.0567**	**0.0179**	**0.0018**	**6.0E-5**
	IT	34	24	23	49	49
	CPU	0.05	0.04	0.05	0.56	0.28
63x63	ε	**0.0601**	**0.0118**	**0.0026**	**2.6E-4**	**4.6E-6**
	IT	127	105	105	213	213
	CPU	1.16	0.96	1.12	13.1	6.4
94x94	ε	**0.0413**	**0.0053**	**9.6E-4**	**1.1E-4**	**1.6E-6**
	IT	258	226	226	453	453
	CPU	5.5	5.0	5.8	46.4	30.8
141x141	ε	**0.0279**	**0.0024**	**3.3E-4**	**5.3E-5**	**6.0E-7**
	IT	539	492	492	975	975
	CPU	29.3	25.9	30.7	211.5	146.0

4.3 Fully-Developed Turbulent Flow in Plane Channel: Flow Simulation

In the preceding section we considered a sample heat transfer problem in a plane channel with a fully-developed turbulent flow and heat transfer. A steady-state energy equation has been solved in two dimensions. Detailed description of this sample calculation can be found also in the latest publication [11]. Here, let us consider similar sample problem that is related to the calculation of pressure and velocity distribution for the turbulent flow. Again, we consider the fully-developed turbulent flow in a plane channel shown in Fig. 10. In fact, this is one-dimensional problem and exact solution can be easily obtained by integration of the following governing equation for fully-developed flow as follows

$$\frac{\partial}{\partial y}\left((\nu + \nu_{turb})\frac{\partial W}{\partial y} \right) = \frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{37}$$

where W is the axial component of the velocity vector; p is the pressure; ρ is the density; ν and ν_{turb} are the viscosity and the turbulent viscosity correspondingly. For the later, experimental dependence of Nijsing and Eifler [13] is used in this report:

$$\frac{\nu_{\text{turb}}}{u^* y^+} = \left[1 - \exp\left(-0.407 \frac{\text{y}}{y^+} \right) \right] \exp\left(-e \frac{\text{y}}{y^+} \right), \tag{38}$$

where y is the distance to the wall,

$$e = \log\left[\frac{1-\exp(-0.407)}{0.07}\right], \qquad y^+ = \frac{yu^*}{\nu},$$

and the friction velocity $u^* = \sqrt{\tau_w/\rho}$

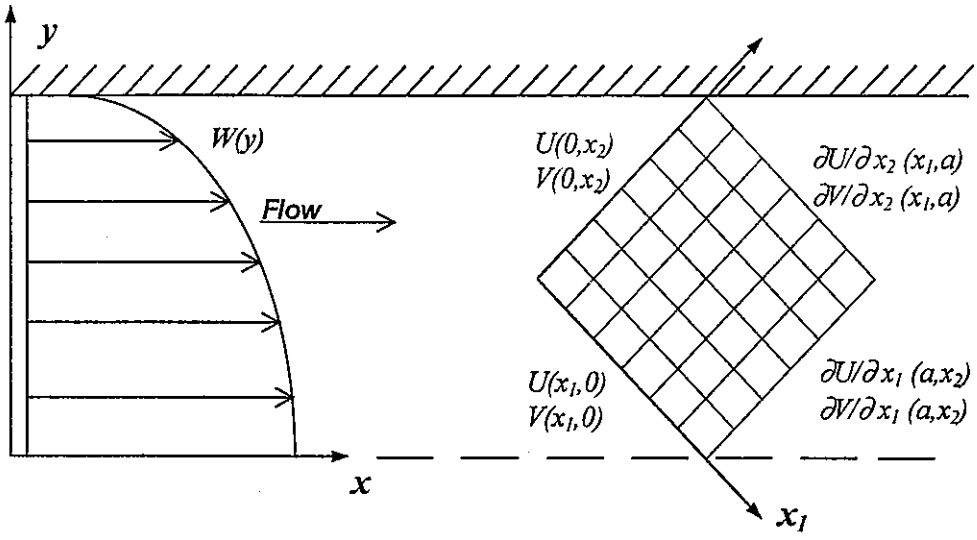

Figure 10. Calculation of the turbulent flow in a plane channel in the Cartesian coordinate system rotated at an angle of 45°

Since the fully-developed flow is considered, the pressure gradient $\frac{\partial p}{\partial x}$ is a constant at any point of the channel and therefore, the exact solution of Eq. (37) can be found by integration over the region $\{0,\ y_{max}\}$. Because of complexity of relation for turbulent viscosity (38), we use numerical integration with 5000 points to obtain a very precise solution for axial velocity profile $W(y)$. Therefore, we will refer this velocity distribution as an "exact" solution to be used for comparison with numerical predictions.

Numerical calculations have been performed for the same two-dimension area shown in Fig. 10 that was used in preceding section. This is a plain square area rotated at the 45° relative to the regular (x,y) Cartesian coordinate system. The following governing equation are used:

i) Momentum Equations for two velocity components

$$U\frac{\partial U}{\partial x_1} + V\frac{\partial U}{\partial x_2} = -\frac{1}{\rho}\frac{\partial p}{\partial x_1} + \frac{\partial}{\partial x_1}\left((\nu + \nu_{turb})\frac{\partial U}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left((\nu + \nu_{turb})\frac{\partial U}{\partial x_2}\right),$$

$$U\frac{\partial V}{\partial x_1} + V\frac{\partial V}{\partial x_2} = -\frac{1}{\rho}\frac{\partial p}{\partial x_2} + \frac{\partial}{\partial x_1}\left((\nu + \nu_{turb})\frac{\partial V}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left((\nu + \nu_{turb})\frac{\partial V}{\partial x_2}\right) \qquad (39)$$

ii) and Continuity Equation:

$$\frac{\partial U}{\partial x_1}+\frac{\partial V}{\partial x_2}=0, \tag{40}$$

where $U(x_1,x_2)$ and $V(x_1,x_2)$ are the components of velocity vectors in x_1 and x_2 directions which are related to the axial velocity profile as follows:

$$U(x_1,x_2)=\cos\left(\frac{\pi}{4}\right)W(y) \quad \text{and} \quad V(x_1,x_2)=\sin\left(\frac{\pi}{4}\right)W(y).$$

Using the exact solution for axial velocity profile $W(y)$ boundary conditions are given as the first type boundary conditions in inlet boundaries

$$\begin{aligned} U(0,x_2)&=\cos\left(\frac{\pi}{4}\right)W(y); \quad V(0,x_2)=\sin\left(\frac{\pi}{4}\right)W(y); \\ U(x_1,0)&=\cos\left(\frac{\pi}{4}\right)W(y); \quad V(x_1,0)=\sin\left(\frac{\pi}{4}\right)W(y); \end{aligned} \tag{41}$$

and the second type boundary conditions in the outlet of the region:

$$\begin{aligned} U(a,x_2)&=\cos\left(\frac{\pi}{4}\right)\frac{\partial W(y)}{\partial x_1}; \quad V(a,x_2)=\sin\left(\frac{\pi}{4}\right)\frac{\partial W(y)}{\partial x_1}; \\ U(x_1,a)&=\cos\left(\frac{\pi}{4}\right)\frac{\partial W(y)}{\partial x_2}; \quad V(x_1,a)=\sin\left(\frac{\pi}{4}\right)\frac{\partial W(y)}{\partial x_2}, \end{aligned} \tag{42}$$

where a is the size of square calculation region as shown in Fig. 10.

The system of equations (39), (40) with boundary conditions (41), (42) is solved numerically using the Upwind, LOAD and EFD discretization schemes. Calculations were performed for the uniform mesh systems with 10x10, 20x20, 40x40, 60x60, 100x100 and 150x150 grid nodes in each direction. The calculated numerical velocity and pressure distributions were compared with exact axial velocity profile $W(y)$ and constant pressure gradient $\frac{\partial p}{\partial x}$.

First, let us compare numerical error in numerically predicted velocity distribution. The results of maximum relative numerical error in velocity defined as

$$\varepsilon_w^{\max}=\underset{i,j}{MAX}\left|\frac{W(y)-\sqrt{U\left(x_1^i,x_2^j\right)^2+V\left(x_1^i,x_2^j\right)^2}}{W(y)}\right|, \tag{43}$$

where $U\left(x_1^i,x_2^j\right)$ and $V\left(x_1^i,x_2^j\right)$ are numerically predicted velocity vectors are shown in Fig. 11. One can see that the EFD scheme predicts the best solution (i.e. the minimum numerical error) for the almost every mesh size except the case of 10x10 grid nodes. Surprisingly, for the low number of grid nodes numerical error in the upwind and LOAD schemes grows with the number of nodes. However, if one considers that it is a *maximum relative* error given by Eq. (43) those results can be explained by the fact that worst numerical prediction in velocity often occurs near the wall. Therefore, the bigger number of grid points for uniform mesh system is provided, the first point is close to the wall and the bigger numerical error can be expected for this point. Nevertheless, the decline in maximum relative error in velocity distribution is monotonous in case of the EFD

scheme. It should be noted here, that all methods considered could predict quite accurate numerical solution of velocity distributions and numerical error *averaged* over the calculation area is lower than 1% for the all cases. This result is expectable because boundary conditions are given for velocity values by the exact solution and velocity distribution has no enough room to develop inside the calculation area.

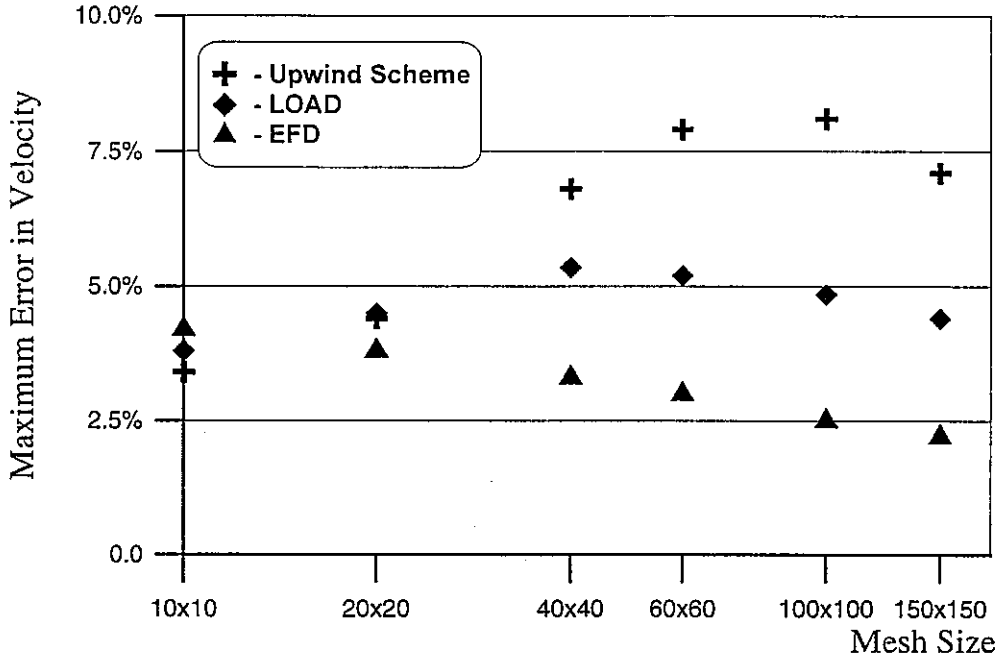

Figure 11. Maximum relative numerical error in velocity distribution vs. number of grid nodes; Re = 10000

Now, let us have a look for more interesting (as well as more important) results of numerical prediction of *pressure gradient*. As it has been mentioned, results for velocity distribution are within a narrow margin of numerical error because of the nature of boundary conditions. From the other hand, the pressure gradient is not given by boundary conditions directly and it is calculated numerically by solving momentum equations (39) and continuity equation (40). Taking in account that an exact pressure gradient is a constant inside the calculation area, we define an average relative numerical error in pressure distribution as follows:

$$\bar{\varepsilon}_p = \frac{1}{N_{points}} \sum_{i,j} \left| \frac{\left.\frac{\partial p}{\partial x}\right|_{exact} - \frac{p\left(x_1^i, x_2^j\right) - p\left(x_1^{i+1}, x_2^{j+1}\right)}{\sqrt{\Delta x_1^2 + \Delta x_2^2}}}{\left.\frac{\partial p}{\partial x}\right|_{exact}} \right|, \tag{44}$$

where N_{points} is the total number of evaluated grid nodes and $p\left(x_1^i, x_2^j\right)$ is the calculated value of pressure at the given grid node. Results of calculation of the average relative error in pressure gradient are shown in Fig. 12.

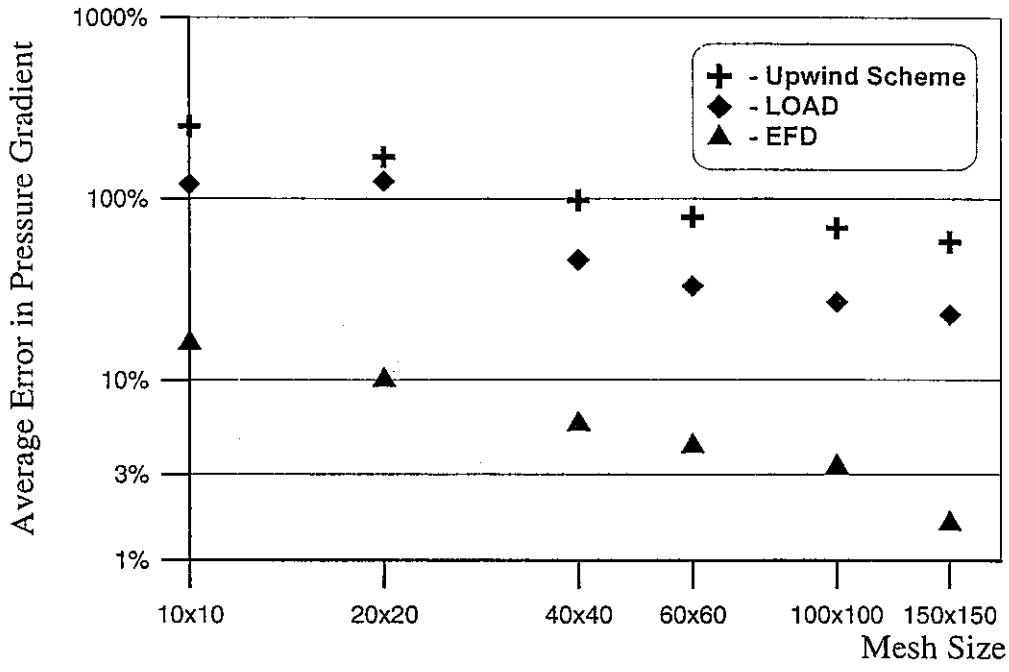

Figure 12. Average relative numerical error in pressure gradient vs. number of grid nodes; Re = 10000

One can see that numerical error in the pressure gradient calculated with the upwind and LOAD schemes is very high even for very fine mesh system with 150x150=22500 grid nodes in total. However, the use of EFD scheme makes it possible to reach a satisfactory agreement with exact pressure gradient even for rough meshes. The huge error in pressure gradient for upwind and LOAD scheme can be explained by using uniform grid that does not allow fine meshing near the wall. It is well known that for turbulent flow calculations it is necessary to place at least a few grid nodes at the boundary layer region, otherwise an accurate numerical solution cannot be guaranteed. In our case the boundary conditions are given by velocity and pressure distribution is calculated by flow development. That is why the error in pressure gradient is very high even the velocity values are calculated with good accuracy. From the other hand, if pressure gradient is given as a boundary condition, one can expect good numerical results in pressure but velocity distribution.

Let us consider why the EFD scheme performs much better (in orders of magnitude of numerical error in pressure) that the regular upwind and even advanced LOAD schemes. From our point of view the reason is that EFD takes into account the distribution of diffusivity coefficient (turbulent viscosity in this case) within the control volume. Following from the definition of the EFD scheme, it allows a better prediction of interface fluxes (shear stresses in this case) between control volumes. Finally, it results in high accuracy of numerical method.

It should be noted that for laminar flow calculations the difference between LOAD and EFD is not so significant but also notable.

5. CONCLUSIONS

In this report, the EFD (Efficient Finite Differencing) scheme has been developed originally to solve convection-diffusion transport phenomena for nuclear reactor problems. These phenomena are pertaining to the fluid flow and heat transfer in nuclear reactor fuel assemblies, steam generators, heat exchangers, containments, ..., etc. Frequently, transient problems must be solved in three-dimensions in order to predict the flow and temperature distribution correctly. Consequently, one cannot use a two-dimensional approximation because characteristics of the transport phenomena are essentially three-dimensional. As a result, the 3D transient analysis of nuclear reactor thermal hydraulics requires a lot of computer memory and CPU time and therefore any increase in efficiency is desirable and urgent to reduce the cost of calculations. Moreover, real-time simulators require that computations per each time step must be completed in given time and accuracy of the simulation depends on the order of a numerical error in a velocity and a temperature distribution.

The main benefit of EFD is the ability to predict more accurate numerical solution on the coarse grid. The discretization with EFD are performed by using one-dimensional locally "exact" analytical solutions for each interface between control volumes with two main assumptions:

- The distribution of the diffusion coefficient between two nodes can be approximated by linear interpolation
- The source term, which is a sum of the physical source term and partial derivatives from the other directions including the time-derivative term, is also linear between two grid nodes

Using these assumptions, the EFD discretization has been derived. When applied to the convection-diffusion transport equation, it results in a linear equation with standard three-point links in each direction. Therefore, in two dimensions, it generates a regular five-point discretization while a seven-point equation is produced in three dimensions.

As a disadvantage, the necessity of iteration procedure to correct an extra-source term should be mentioned. Another inconvenience is the complicated discretization formula that takes about three times more CPU time per node than the standard upwind scheme.

However, the numerical accuracy with EFD is a few orders of magnitude better than that with standard schemes on the same grid. On the other hand, in order to attain the same given accuracy, the number of grid nodes is significantly smaller. This means the CPU time demanded to meet that accuracy reduces sharply in the case of EFD.

The EFD scheme has been compared with the exponential differencing scheme of Spalding, the LOAD scheme of Wong and Raithby, the QUICK scheme of Leonard and the LENS scheme of Sakai. Benchmark calculations have been carried out including one-dimensional steady-state and transient flows as well as simplified two-dimensional problems of turbulent fluid flow and heat transfer in a plane channel. For the fixed number of grid points, EFD has shown significantly better accuracy than the other finite-difference methods investigated.

We believe also, that an additional research would be useful to prove the stability of the EFD scheme with different numerical experiments by applying this scheme to more realistic and practical problems. We expect that EFD will show good performance for most turbulent flow simulations. Nevertheless, stability and convergence of this scheme should be proven for fluid flow calculations.

REFERENCES

1. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, D.C., (1980).
2. B.P. Leonard, "A Stable and Accurate Convective Modeling Procedure Based on Quadratic Upstream Interpolation", *Computer Methods in Applied Mech. and Eng.*, Vol. 19, pp. 59-98 (1979).
3. H.H. Wong and G.D. Raithby, "Improved Finite-Difference Methods Based on a Critical Evaluation on the Approximation Errors", *Numerical Heat Transfer*, vol. 2, pp. 139-163 (1979).
4. K.C. Karki, P.S. Sathyamurthy, S.V.Patankar, Perfomance of Multigrid Method with an Improved Discretization Scheme for Three-Dimensional Fluid Flow Calculations, *Numerical Heat Transfer, Part B*, vol. 29, pp. 275-288 (1996).
5. K. Sakai, A Locally Exact Numerical Scheme with Nonoscillation Properties for Stationary Transport Equations with Absorption and Source Terms, *Nuclear Science and Eng.*, vol. 123, pp. 57-67 (1996).
6. C. Prakash, Application of the Locally Analytic Differencing Scheme to Some Test Problems for the Convection-Diffusion Equation, *Numerical Heat Transfer*, vol. 7, pp. 165-182 (1984).
7. V. Kriventsev, "A New Efficient Finite-Difference Discretization of Convection-Diffusion Equations", *Proc. Annual Meeting on Nuclear Technology'96*, Mannheim, pp. 130-133 (1996).
8. V. Kriventsev, H. Ninokata, "An Application of Efficient Finite-Differencing Scheme to Transient Problems of Fluid Flow and Heat Transfer", *Proc. 5th Int. Conf. On Nuclear Engineering ICONE-5,* Nice, France (1997).
9. K.C. Karki, S.V. Patankar, H.C. Mongia, Three-Dimensional Fluid Flow Calculations Using a Flux-Spline Method, *AIAA Journal,* Vol. 28, No. 4, pp. 631-634 (1990).
10. V. Kriventsev and H. Ninokata, *Proc. 8th Int. Topical Meeting on Nuclear Reactor Thermal-Hydraulics (NURETH-8)*, Kyoto, 1, p. 572 (1997).
11. V. Kriventsev, H. Ninokata, An Effective Locally-Exact Finite-Difference Scheme For Convection-Diffusion Problems, *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 36, No 2, (1999).
12. V. Bets, V. Kriventsev, An Object-Oriented Approach In Problems of Numerical Heat and Mass Transfer, *Proc. 5th Int. Conf. On Nuclear Engineering ICONE-5,* Nice, France (1997).
13. R. Nijsing, and W. Eifler. *Progress in Heat and Mass Transfer,* Vol. 7, p. 115 (1971).