Chapter 6

Petrov-Galerkin Formulations for Advection Diffusion Equation

In this chapter we'll demonstrate the difficulties that arise when GFEM is used for advection (convection) dominated problems. Several cures will be suggested such as the use of upwinding, artificial diffusion, Petrov-Galerkin formulations and stabilization techniques.

6.1 GFEM for 1D Advection Diffusion Equation using Linear Elements

Consider the following 1D, steady AD equation

$$U\frac{dT}{dx} - \nu\frac{d^2T}{dx^2} = f \qquad \text{in} \quad [0, L] \tag{6.1}$$

$$T(0) = 0$$

$$T(L) = 0$$

where $T(x)$ is the scalar unknown, f is the known force function, U and ν are the known constant velocity and diffusivity, respectively. GFEM formulation yields the following elemental stiffness matrix, written as the summation of elemental convection matrix C^e and elemental diffusion matrix D^e.

$$K_{ij}^e = C_{ij}^e + D_{ij}^e = \int_{\Omega^e} U\, S_i \frac{dS_j}{dx}\, dx + \int_{\Omega^e} \nu \frac{dS_i}{dx}\frac{dS_j}{dx}\, dx \tag{6.2}$$

Using linear elements and considering constant U and ν, C^e and D^e matrices can be calculated as

$$[C^e] = U\int_{\Omega^e} \begin{bmatrix} S_1\frac{dS_1}{dx} & S_1\frac{dS_2}{dx} \\ S_2\frac{dS_1}{dx} & S_2\frac{dS_2}{dx} \end{bmatrix} dx = \frac{U}{2}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \tag{6.3}$$

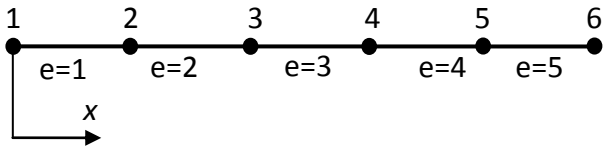
$$[D^e] = \nu\int_{\Omega^e} \begin{bmatrix} \frac{dS_1}{dx}\frac{dS_1}{dx} & \frac{dS_1}{dx}\frac{dS_2}{dx} \\ \frac{dS_2}{dx}\frac{dS_1}{dx} & \frac{dS_2}{dx}\frac{dS_2}{dx} \end{bmatrix} dx = \frac{\nu}{h^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6.4}$$

As seen the contribution of the diffusion term to $[K^e]$ is symmetric, whereas the contribution of the advection term is not.

Considering the force function f to be constant, elemental force vector becomes

$$F^e = f\int_{\Omega^e} \begin{Bmatrix} S_1 \\ S_2 \end{Bmatrix} dx = \frac{fh^e}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{6.5}$$

Consider a mesh of 5 equi-length linear elements as shown below



The assembled system of equations will be

$$\left[\frac{U}{2}\begin{bmatrix}-1 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & -1 & 0 & 1 & \\ & & & -1 & 0 & 1 \\ & & & & -1 & -1\end{bmatrix}+\frac{\nu}{h^e}\begin{bmatrix}1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & -1\end{bmatrix}\right]\begin{Bmatrix}T_1\\T_2\\T_3\\T_4\\T_5\\T_6\end{Bmatrix}=\frac{fh^e}{2}\begin{Bmatrix}1\\2\\2\\2\\2\\1\end{Bmatrix}$$

Therefore the algebraic equation for an interior node n is

$$\frac{U}{2}(T_{n+1}-T_{n-1})-\frac{\nu}{h^e}(T_{n+1}-2T_n+T_{n-1})=fh^e \tag{6.6}$$

Dividing the equation by h^e we get

$$U\left(\frac{T_{n+1}-T_{n-1}}{2h^e}\right)-\nu\left(\frac{T_{n+1}-2T_n+T_{n-1}}{(h^e)^2}\right)=f \tag{6.7}$$

6.2 Finite Difference Discretization of 1D Advection Diffusion Equation

It is also possible to discretize 1D, steady AD equation using a finite difference approach. The following are the second order accurate central differencing based approximate first and second derivatives written for an interior node *n* of a finite difference mesh with a constant node spacing of Δx

$$\left.\frac{dT}{dx}\right|_n\approx\frac{T_{n+1}-T_{n-1}}{2\Delta x}\quad,\qquad\left.\frac{d^2T}{dx^2}\right|_n\approx\frac{T_{n+1}+2T_n-T_{n-1}}{(\Delta x)^2} \tag{6.8}$$

Using these in the DE we get the following discrete equation for node *n*

$$U\left(\frac{T_{n+1}-T_{n-1}}{2\Delta x}\right)-\nu\left(\frac{T_{n+1}-2T_n+T_{n-1}}{(\Delta x)^2}\right)=f \tag{6.9}$$

which is the same as the equation obtained by GFEM.

> It is possible to conclude that GFEM has similarities to the use of central differencing in Finite Difference Method.

6.3 Sample 1D Solutions using GFEM

It is easier to study AD equation by introducing the following elemental Peclet number

$$Pe = \frac{Uh^e}{2\nu} \tag{6.10}$$

which is ratio of convection and diffusion. As the Peclet number gets larger the problem gets more convection dominated.

We'll solve 1D, steady AD equation with $U = 1, f = 1$ on a mesh of 10 equi-length elements. By taking $L = 1$, element lengths are fixed to $h^e = 0.1$. Exact solution is known to be

$$T_{exact} = \frac{1}{U}\left(x - \frac{1 - e^{Ux/\nu}}{1 - e^{U/\nu}}\right) \tag{6.11}$$

GFEM results for different diffusivity values, therefore for different elemental Peclet numbers, are shown in Figure 6.1. As seen, as Pe increases, a sharp gradient, sometimes called a boundary layer, develops at the right end of the domain. For high Pe cases GFEM provides a wiggly solution with spurious node to node oscillations, failing to capture the highly nonlinear change. This oscillatory behavior is seen for $Pe > 1$.

6.4 A Special Solution with Exact Results at the Nodes

The reason of unphysical oscillations seen in the previous section is the truncation errors introduced by the GFEM (or equally the Finite Difference method with central differencing) formulation given by Eqn (6.7) (or equally by Eqn (6.9)). It is possible to write Eqn (6.7) in the following way by using the elemental Peclet number definition

$$\frac{U}{2h^e}\left[\left(1 - \frac{1}{Pe}\right)T_{n+1} + \frac{2}{Pe}T_n - \left(1 + \frac{1}{Pe}\right)T_{n-1}\right] = f \tag{6.12}$$

In this section we'll try to find another very special discretization scheme, similar in structure to GFEM given above, but providing exact solution at each node of a uniform mesh of linear elements for any h^e and Pe values.

Let's say that this special scheme provides the following discrete equation for $f = 1$

$$AT_{n+1} + BT_n + CT_{n-1} = 1 \tag{6.13}$$

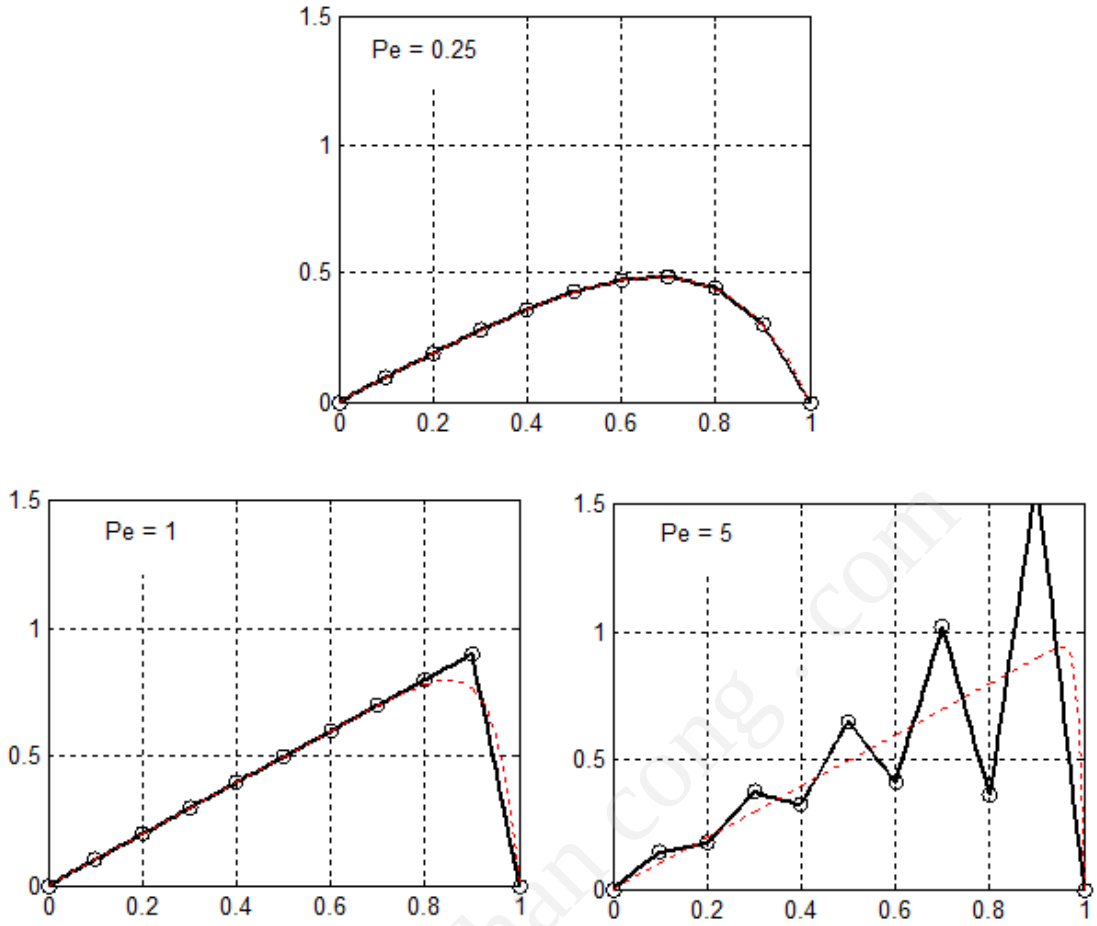


Figure 6.1 GFEM solution of 1D, steady AD equation at three different *Pe* values. Dashed red lines show exact solutions.

We want to find the unknown coefficients A, B and C of this discretization so that it yields exact nodal values at nodes $n-1$, n and $n+1$. Following the details from reference [1], this special discretization is determined as

$$\frac{U}{2h^e}[(1-\coth(Pe))T_{n+1} + 2\coth(Pe)\,T_n - (1+\coth(Pe))T_{n-1}] = 1 \qquad (6.14)$$

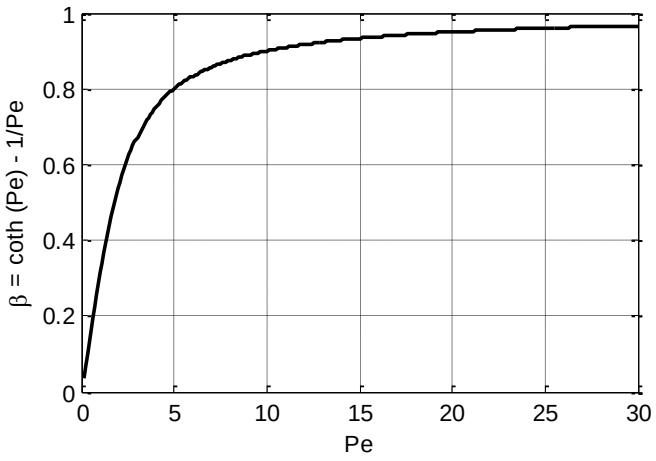
which can be arranged into the following form

$$U\left(\frac{T_{n+1}-T_{n-1}}{2h^e}\right) - (\nu+\bar{\nu})\left(\frac{T_{n+1}-2T_n+T_{n-1}}{(h^e)^2}\right) = 1 \qquad (6.15)$$

where $\bar{\nu}$ is known as artificial (numerical) diffusion (dissipation) given by

$$\bar{\nu} = \frac{\beta U h^e}{2} = \beta \nu Pe \qquad \text{with} \qquad \beta = \coth(Pe) - \frac{1}{Pe} \qquad (6.16)$$

The variation of β with elemental Peclet number is shown below



Note that $\bar{\nu}$ depends only on the parameters of the DE and element length. For positive U values, which is the case discussed here, $\bar{\nu}$ is always positive. Here we assume that the actual diffusivity ν is also positive, which is the case for real problems with a physical background.

If we compare this "nodal-exact discretization" with the one obtained previously for GFEM, i.e. compare Eqns (6.7) and (6.15), we'll notice that the advection terms containing U are the same, but diffusion terms containing ν are different. Artificial diffusion term is missing in GFEM.

> Therefore it is possible to conclude that GFEM introduces a truncation error in the form of diffusion operator. In other words GFEM discretization actually provides a solution for a DE with wrong (less than the actual) diffusivity.

The idea of using artificial diffusivity that will be explained in the coming sections is based on this observation. But this quick solution might not be the best solution, as we will demonstrate. Finally, it is important to note that the nodal-exact solution of this section is valid only if U, ν and f of the AD equation are constants, and it is difficult to obtain such a solution for a more general case.

6.5 Use of Upwind Differencing to Stabilize AD Solutions

Discussion of the previous section is the origin of the techniques developed for improving GFEM solution of AD equation. The idea is to solve a DE with increased diffusion, i.e. solve a DE with an increased diffusivity instead of the physical diffusivity ν given in the DE. This commonly used technique is known in the literature as "use of artificial diffusivity (dissipation)".

In a finite difference formulation the required artificial diffusion can be obtained by using first-order accurate upwind differencing for the first derivative, instead of the second order accurate central differencing used previously in Section 6.2. If the convection is in the positive x direction, i.e. if $U > 0$, first derivative is discretized as

$$\left.\frac{dT}{dx}\right|_n \approx \frac{T_n - T_{n-1}}{\Delta x} \tag{6.17}$$

Using this new approximate first derivative and the previously used central differencing for the second derivative the following discrete equation is obtained for an internal node n of the mesh

$$U\left(\frac{T_n - T_{n-1}}{\Delta x}\right) - \nu\left(\frac{T_{n+1} - 2T_n + T_{n-1}}{(\Delta x)^2}\right) = f \tag{6.18}$$

which can be put into the following form

$$U\left(\frac{T_{n+1} - T_{n-1}}{2\Delta x}\right) - \left(\nu + \frac{U\Delta x}{2}\right)\left(\frac{T_{n+1} - 2T_n + T_{n-1}}{(\Delta x)^2}\right) = f \tag{6.19}$$

This equation is very similar to Eqn (6.9) obtained previously in Section 6.2 using central differencing for both first and second derivatives. The difference is the increase of diffusion from ν to $\nu + \frac{U\Delta x}{2}$.

Therefore the following two finite difference formulations provide the same approximate solution

1. Use upwind differencing for the first order derivative and using a diffusivity of ν
2. Use central differencing for both derivatives with an increased diffusivity of $\nu + \frac{Uh^e}{2}$

The above formulation is known as full upwinding, which corresponds to the use of a constant upwinding parameter of $\beta = 1$, instead of using a Pe dependent β as in Eqn (6.16).

In Section 6.2 we showed that GFEM is numerically the same as a finite difference discretization with both first and second order derivatives approximated using central differences. Therefore it is possible to test the performance of upwind differencing using a GFEM solver, in which the effect of upwinding is simulated by using artificial dissipation. The results of such a test are shown in Figure 6.2. This figure should be compared with Figure 6.1 to see the effect of artificial diffusion.

As demonstrated in these solutions use of upwind differencing for the convective term results in stable solutions (no unphysical oscillations for high Pe cases). But this stability comes with a price; these solutions turn out to be excessively dissipative (overly diffusive). This is the major concern of upwind differencing and it is heavily criticized in the literature for this. Since upwind differencing is numerically the same as using central differencing with artificial diffusion, the phrase “artificial diffusion” also took its share of criticisms.

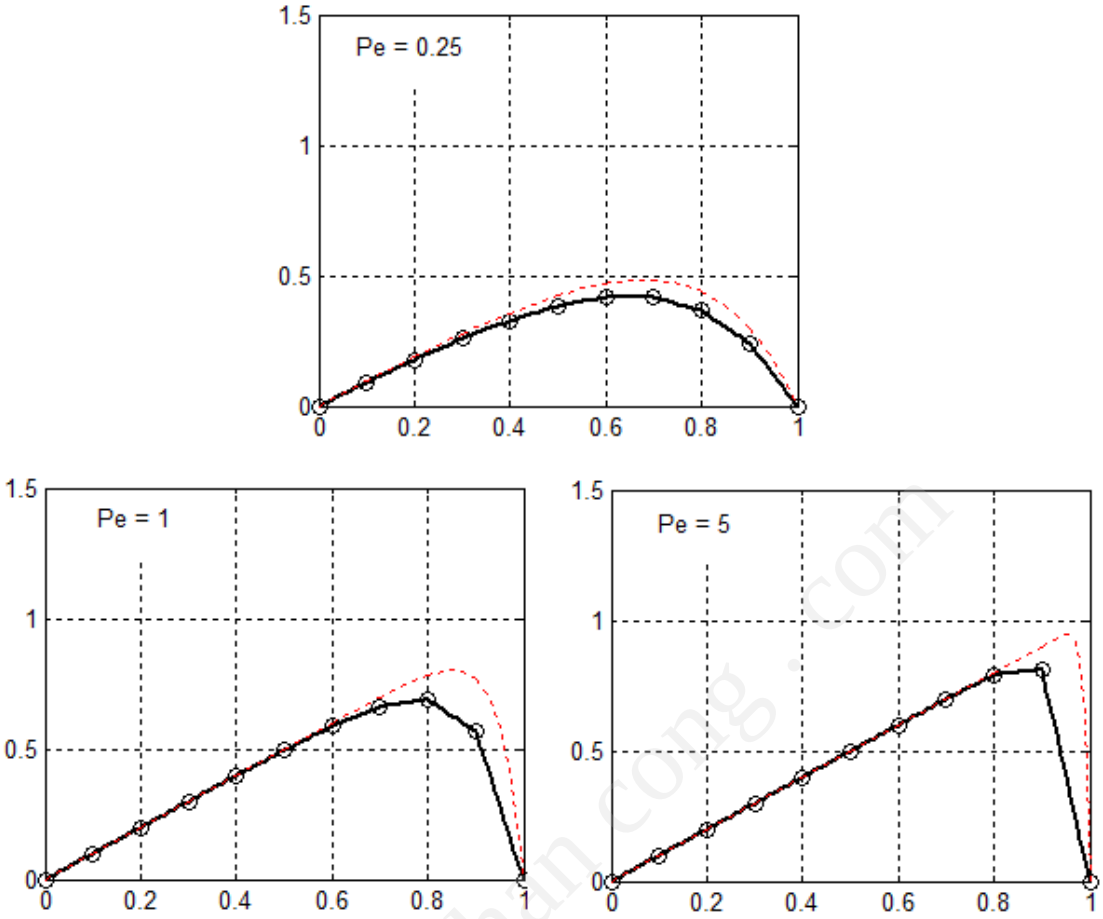
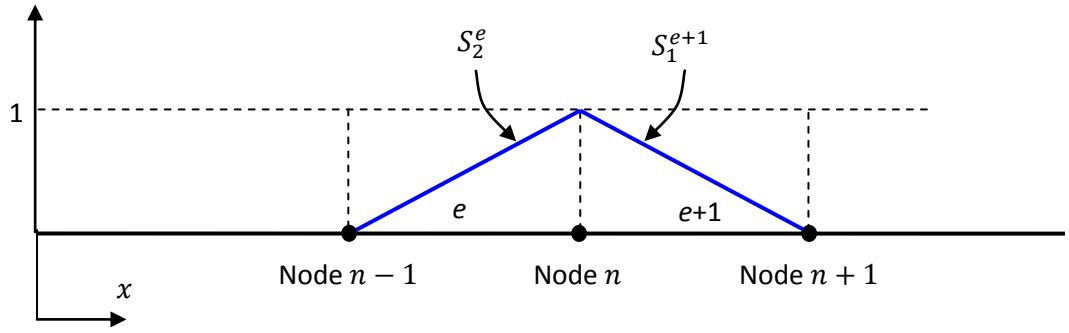


Figure 6.2 Performance of GFEM with artificial diffusion (corresponding to full upwinding, $\beta = 1$) at three different Pe values. Compare the results with those of Figure 6.1.

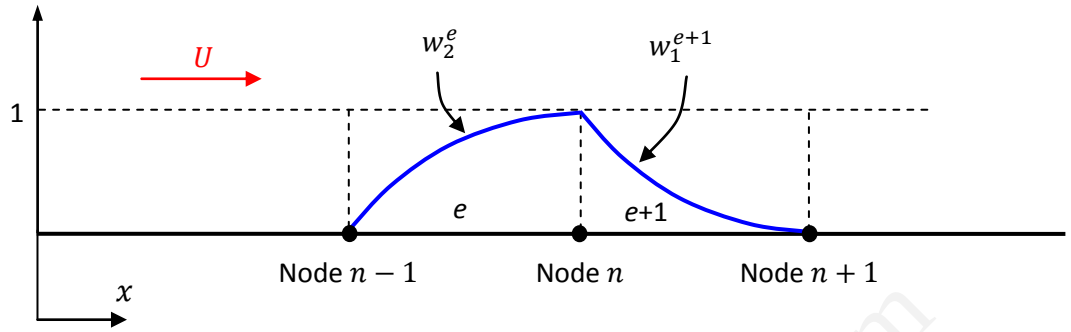
6.6 Upwind Type Finite Elements – A Petrov Galerkin Formulation

The upwind effect used in finite difference can be achieved in finite elements by using Petrov-Galerkin formulations (Petrov GFEM) instead of GFEM, i.e. we use weight functions that are different than the shape functions used for unknown approximation.

Consider the following one dimensional mesh of equi-length linear elements.



Let's concentrate on node n. Only the shown two shape functions contribute to the equation written for node n. If $U > 0$, node $n-1$ is said to be on the upstream side of node n, and node $n+1$ is on the downstream side of node n. In Petrov GFEM, instead of selecting the weight functions to be the same as the shown shape functions, we distort them to the upwind side as shown below.



This kind of distorted weight functions can be obtained by adding bubble functions (they have zero values on the nodes and they are nonzero on elements' interiors, so when plotted they look like bubbles over the elements) to the original linear shape functions. One possibility is to use

$$w_2^e = \frac{1}{2}(1+\xi) + \frac{3}{4}\beta(1-\xi^2) \tag{6.20}$$

$$w_1^{e+1} = \frac{1}{2}(1-\xi) - \frac{3}{4}\beta(1-\xi^2) \tag{6.21}$$

where β is a parameter that controls the amount of upwinding we want to use. $\beta = 0$ corresponds to no upwinding, which corresponds to GFEM.

Elemental stiffness matrices using these modified weight functions are

$$C^e = U\int_{\Omega^e}\begin{bmatrix} w_1\frac{dS_1}{dx} & w_1\frac{dS_2}{dx} \\ w_2\frac{dS_1}{dx} & w_2\frac{dS_2}{dx}\end{bmatrix}dx = \frac{U}{2}\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix} + \frac{U}{2}\begin{bmatrix}\beta & -\beta\\ -\beta & \beta\end{bmatrix} \tag{6.22}$$

$$D^e = \nu\int_{\Omega^e}\begin{bmatrix} \frac{dw_1}{dx}\frac{dS_1}{dx} & \frac{dw_1}{dx}\frac{dS_2}{dx} \\ \frac{dw_2}{dx}\frac{dS_1}{dx} & \frac{dw_2}{dx}\frac{dS_2}{dx}\end{bmatrix}dx = \frac{\nu}{h^e}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \tag{6.23}$$

Comparing these with Eqns (6.3) and (6.4) obtained using GFEM, it is seen that modifying the weight functions using bubble functions changes the convection matrix, but not the diffusion matrix. However, extra terms of the convection matrix, shown in red, are similar in structure to the diffusion matrix and the overall stiffness matrix can also be tought of the following way

$$K^e = C^e + D^e = \frac{U}{2}\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix} + \frac{\nu + U\beta h^e/2}{h^e}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \tag{6.24}$$

In this form it is clearly seen that modifying weight functions using bubble functions to get an upwinding effect is numerically similar to using artificial diffusion. The amount of artificial diffusion is $\bar{\nu} = U\beta h^e/2$, which is the same as the one given in Eqn (6.16).

Test results obtained with this formulation for $Pe = 5$ with different β values are shown in Figure 6.3. As seen, by fine tuning the upwinding parameter β, it is possible to obtain stable results. The optimum value of $\beta = \coth(Pe) - 1/Pe$, which is not tried here, should provide a solution that is exact at the mesh nodes. As seen in Figure 6.3, similar to the previous upwind solutions, these solutions are also excessively diffusive if the upwinding parameter β is used more than enough. Proper selection of β is problem and mesh dependent.

It is now possible to make the following important conclusion, similar to the one made in Section 6.5.

The following two finite element formulations provide the same approximate solution

1. Petrov GFEM with weight functions modified using bubble functions.
2. GFEM with an increased diffusivity of $\nu + U\beta h^e/2$ (or artificial diffusivity of $\bar{\nu} = U\beta h^e/2$)

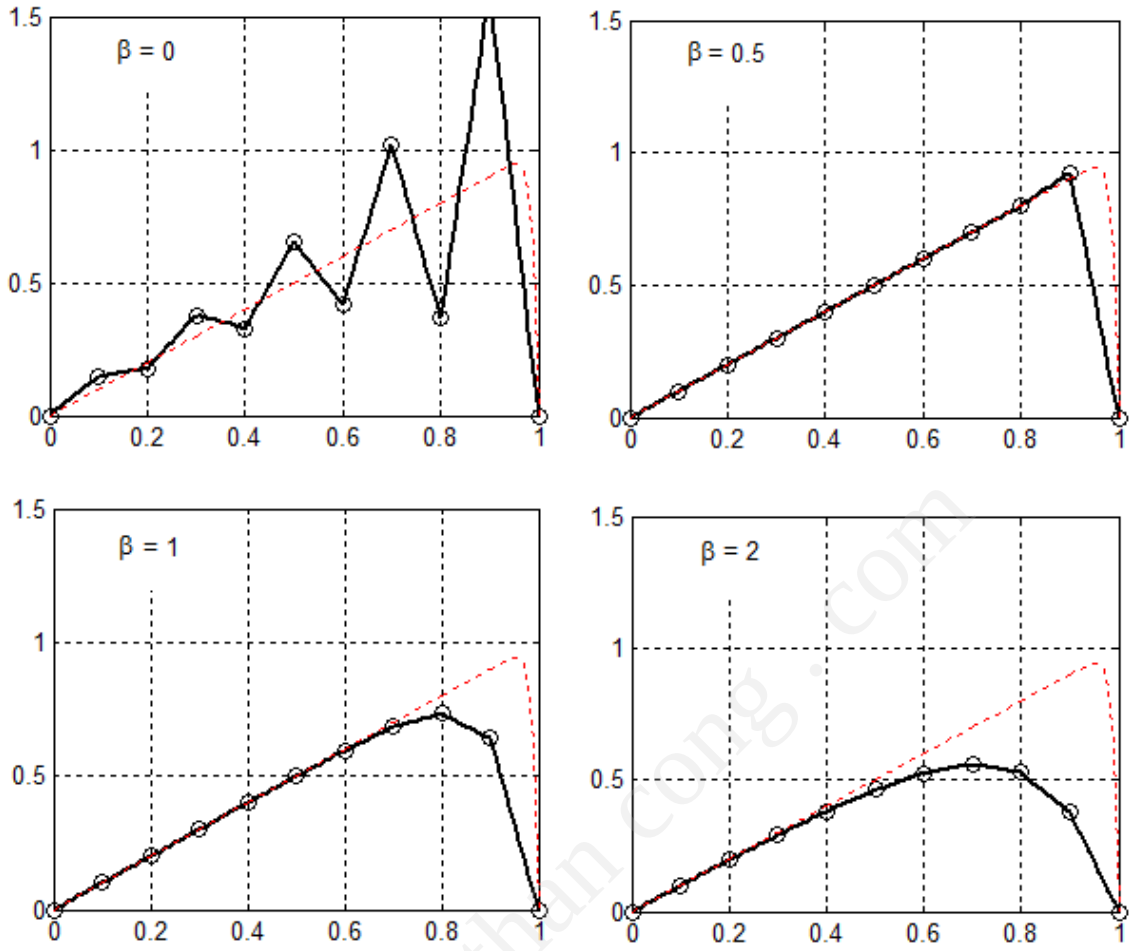
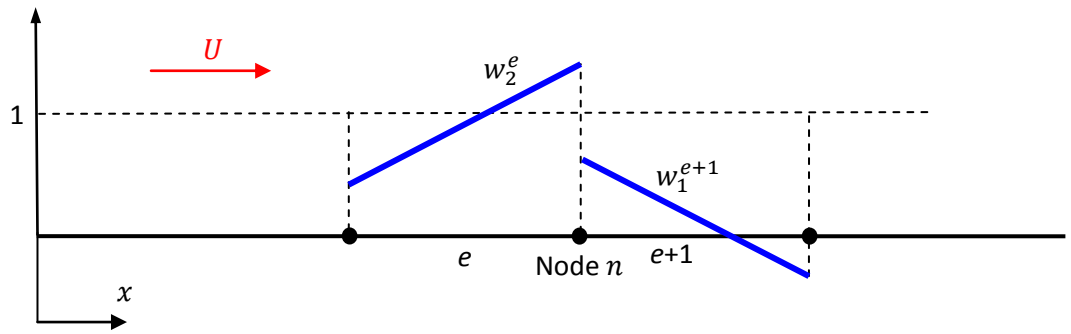


Figure 6.3 Performance of Petrov GFEM with bubble functions with four different upwinding parameters. All results are for $Pe = 5$.

6.7 An Alternative Petrov Galerkin Formulation – Inconsistent Streamline Upwind (SU)

In the previous section a Petrov GFEM formulation is presented based on weight functions modified using bubble functions. It is also shown that this version of Petrov GFEM is numerically the same as using GFEM with artificial diffusion. Use of bubble functions is not the only way of obtaining a Petrov GFEM formulation. A simpler alternative is shown below.



As seen the weight functions are obtained by shifting the shape functions up and down according to the flow direction. For positive U values (convection is to the right) weight functions of the elements can be written as

$$w_i^e = S_i^e + \beta \frac{dS_i^e}{d\xi} \tag{6.25}$$

where β is a parameter that controls the amount of up and down shifts ($\beta = 0$ coresponds to GFEM). For linear elements, slopes of the shape functions are $\frac{dS_1^e}{d\xi} = -0.5$ and $\frac{dS_2^e}{d\xi} = 0.5$ and using these values in Eqn (6.26) modified weight functions become

$$w_1^{e+1} = \frac{1}{2}(1-\xi) - \frac{\beta}{2} \quad , \quad w_2^e = \frac{1}{2}(1+\xi) + \frac{\beta}{2} \tag{6.26}$$

Using these modified weight functions to evaluate both the convection and diffusion parts of the stiffness matrix is known as consistent Petrov GFEM. This is what we did in the previous section with bubble functions, i.e. we used modified weight functions in Eqns (6.22) and (6.23). However, it is also possible to use modified functions only to evaluate the convection term, which is known as inconsistent Petrov GFEM. This second choice provides the following results

$$C^e = U \int_{\Omega^e} \begin{bmatrix} w_1 \frac{dS_1}{dx} & w_1 \frac{dS_2}{dx} \\ w_2 \frac{dS_1}{dx} & w_2 \frac{dS_2}{dx} \end{bmatrix} dx = \frac{U}{2}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} + \frac{U}{2}\begin{bmatrix} \beta & -\beta \\ -\beta & \beta \end{bmatrix} \tag{6.27}$$

$$D^e = \nu \int_{\Omega^e} \begin{bmatrix} \frac{dS_1}{dx}\frac{dS_1}{dx} & \frac{dS_1}{dx}\frac{dS_2}{dx} \\ \frac{dS_2}{dx}\frac{dS_1}{dx} & \frac{dS_2}{dx}\frac{dS_2}{dx} \end{bmatrix} dx = \frac{\nu}{h^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6.28}$$

which are the same as the ones obtained in the previous section using bubble functions. In Exercise 6.1 you are asked to work on the consistent version of this formulation.

Obtaining an upwind effect by modifying shape functions as discussed in this section is known as streamline upwinding (SU) in FEM. The word "streamline" corresponds to the fact that weight functions are modified according to the flow direction, which is dictated by the sign of U in 1D. The formulation described in this section will later be referred as inconsistent SU.

6.8 Inconsistent Streamline Upwinding (SU) in 2D

Two-dimensional AD equation with constant diffusivity ν is given by

$$\vec{V} \cdot \nabla T - \nabla \cdot (\nu \nabla T) = f \tag{6.29}$$

Streamline upwinding idea of the previous section can be extended to 2D such that weight functions are modified in such a way that the necessary artificial diffusion is added only in the flow direction,

not perpendicular to it, which is known in literature as "avoiding crosswind diffusion". In order to achieve this, the following artificial diffusion matrix can be used

$$\tilde{\nu}_{ij} = \bar{\nu}\frac{V_i V_j}{|\vec{V}|^2} \tag{6.30}$$

where $\vec{V}$ is the known 2D velocity vector. Value of $\bar{\nu}$ determines the amount of added artificial diffusion and its determination is not straightforward for a 2D problem. One possibility for quadrilateral elements is to extend what we did in 1D with Eqn (6.16) to 2D as follows

$$\bar{\nu} = \frac{1}{2}\left(\bar{\xi}\, V_\xi\, h_\xi + \bar{\eta}\, V_\eta\, h_\eta\right) \tag{6.31}$$

where

$$\bar{\xi} = \coth\left(Pe_\xi\right) - 1/Pe_\xi \quad , \quad \bar{\eta} = \coth\left(Pe_\eta\right) - 1/Pe_\eta$$

$$Pe_\xi = V_\xi h_\xi / 2\nu \quad , \quad Pe_\eta = V_\eta h_\eta / 2\nu$$

$$V_\xi = \overrightarrow{e_\xi} \cdot \vec{V} \quad , \quad V_\eta = \overrightarrow{e_\eta} \cdot \vec{V}$$

where the distances h_ξ and h_η and the directions $\overrightarrow{e_\xi}$ and $\overrightarrow{e_\eta}$ are as shown in Figure 6.4. ξ and η directions are obtained by joining the mid points of faces 2 and 4, and 1 and 3, respectively. Note that these directions are not the same as the coordinates used for a master element, although same symbols are used for both.

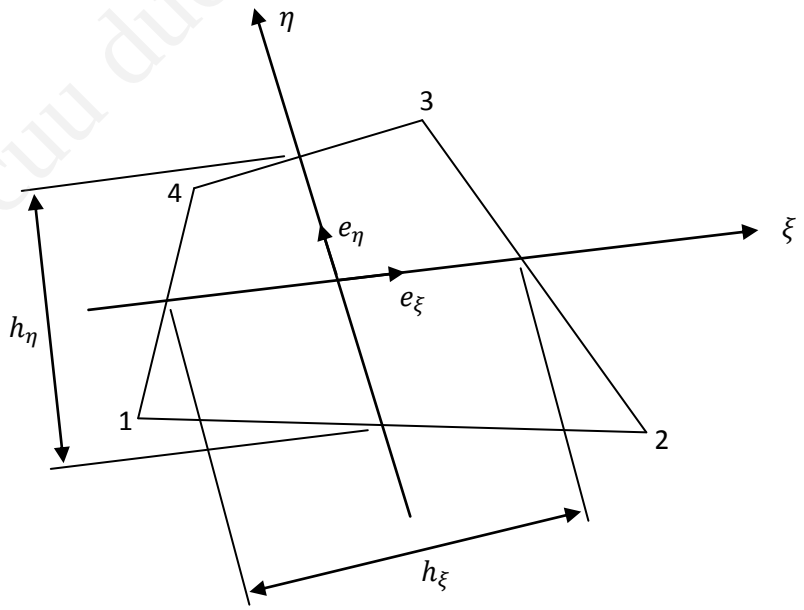


Figure 6.4 Details of SU formulation for a 2D quadrilateral element

As previously derived in Chapter 2, stiffness matrix of the 2D AD equation with constant ν is

$$K^e = C^e + D^e = \int_{\Omega^e} w\,\vec{V}\cdot\nabla T\,d\Omega + \int_{\Omega^e} \nabla w\cdot(\nu\,\nabla T)\,d\Omega \tag{6.32}$$

To obtain 2D streamline upwind (SU) formulation artificial diffusion in matrix form is added to the already existing diffusion

$$K^e = \int_{\Omega^e} w\,\vec{V}\cdot\nabla T\,d\Omega + \int_{\Omega^e} \nabla w\cdot\left[\left(\nu\bar{\bar{I}} + \tilde{\boldsymbol{\nu}}\right)\nabla T\right]d\Omega \tag{6.33}$$

where the 2x2 identity matrix $\bar{\bar{I}}$ is introduced to have a proper matrix summation. The above equation can arranged as

$$K^e = \underbrace{\int_{\Omega^e} w\,\vec{V}\cdot\nabla T\,d\Omega + \int_{\Omega^e} \nabla w\cdot(\nu\,\nabla T)\,d\Omega}_{\text{GFEM contribution}} + \underbrace{\int_{\Omega^e} \nabla w\cdot(\tilde{\boldsymbol{\nu}}\,\nabla T)\,d\Omega}_{\text{SU contribution}} \tag{6.34}$$

Using the definition of $\tilde{\nu}$ matrix given by Eqn (6.30), SU contribution part can be written as follows

$$\text{SU contribution} = \int_{\Omega^e} \frac{\bar{\nu}}{|\vec{V}|^2}\left(\vec{V}\cdot\nabla w\right)\left(\vec{V}\cdot\nabla T\right)d\Omega \tag{6.35}$$

This SU contribution term can be combined with the original C^e term of GFEM to get

$$K^e = \int_{\Omega^e}\left[w + \frac{\bar{\nu}}{|\vec{V}|^2}\left(\vec{V}\cdot\nabla w\right)\right]\vec{V}\cdot\nabla T\,d\Omega + \int_{\Omega^e}\nabla w\cdot(\nu\,\nabla T)\,d\Omega \tag{6.36}$$

In this final form it is possible to see that

> Upwinding technique described in this section is an inconsistent Petrov GFEM, which uses the following modified weight function only to evaluate the convection matrix
>
> $$w + \frac{\bar{\nu}}{|\vec{V}|^2}\left(\vec{V}\cdot\nabla w\right)$$

It is possible to show that the weight function modification given above reduces to the form given in Sections 6.6 and 6.7 for the 1D case.

6.9 Generalized Consistent Petrov Galerkin Formulations (Stabilized Formulations)

In literature consistent Petrov Galerkin formulations that are used to obtain non-oscillatory solutions for convection dominated problems are known as stabilized formulations. It is possible to generalize many different stabilized formulations in a single compact form, as shown below.

Consider the 2D AD equation given in Eqn (6.31). Its residual is given as

$$R = \mathcal{L}(T) - f \tag{6.39}$$

where $\mathcal{L}(\ )$ is the following differential operator

$$\mathcal{L}(\ ) = \vec{V} \cdot \nabla(\ ) - \nabla \cdot [\nu\, \nabla(\ )] \tag{6.40}$$

Consistent Petrov GFEM stabilizations can be generalized as the following elemental weak form

$$\int_{\Omega^e} \left[w\, \vec{V} \cdot \nabla T + \nabla w \cdot (\nu \nabla T) \right] d\Omega \quad + \underbrace{\int_{\Omega^e} \mathcal{P}(w)\, \tau\, R(T)\, d\Omega}_{\text{Stabilization term}} = \int_{\Omega^e} f\, w\, d\Omega \ + \int_{\Gamma^e} w\, SV\, d\Gamma \tag{6.41}$$

where a residual based stabilization term is added to the standard GFEM formulation. If GFEM discretization is already providing a successful solution on a certain mesh, i.e. if the residual (error) is low, the effect of the stabilization term will be small. τ is a user selected stabilization parameter with no unique definition. Different selections of $\mathcal{P}(\ )$ operator results in different stabilized formulations, two of which will be discussed in the coming sections.

Note that unlike inconsistent SU formulation of the previous section, consistent Petrov GFEM stabilization given in this section not only modifies the convective part of K^e matrix, but also the diffusive part of it, as well as the F^e vector.

6.9.1 Streamline Upwind Petrov Galerkin (SUPG) Stabilization

One of the most popular consistent Petrov GFEM stabilization is known as Streamline Upwind Petrov Galerkin (SUPG) for which $\mathcal{P}(w)$ and τ are selected as

$$\text{SUPG}: \qquad \mathcal{P}(w) = \vec{V} \cdot \nabla w \qquad , \qquad \tau = \frac{\bar{\nu}}{\left|\vec{V}\right|^2} \tag{6.42}$$

where $\bar{\nu}$ is previously defined for 1D and 2D problems in Sections 6.4 and 6.8, respectively. Alternative selections of τ can be found in reference [1].

Let's try to obtain K^e and F^e for 1D AD equation with constant U and ν. For 1D equation with constant U velocity SUPG formulation uses the following information

$$\mathcal{P}(w) = U\frac{dw}{dx} \quad , \quad \tau = \frac{\bar{\nu}}{U^2} = \frac{\beta U h^e/2}{U^2} = \frac{\beta h^e}{2U} \quad , \quad R(T) = U\frac{dT}{dx} - \nu\frac{d^2T}{dx^2} - f \tag{6.43}$$

Using these in Eqn (6.41) we get

$$\int_{\Omega^e}\left(Uw\frac{dT}{dx} + \nu\frac{dw}{dx}\frac{dT}{dx}\right)d\Omega + \int_{\Omega^e} U\frac{dw}{dx}\frac{\beta h^e}{2U}\left(U\frac{dT}{dx} - \nu\frac{d^2T}{dx^2} - f\right)d\Omega = \int_{\Omega^e} f\,w\,d\Omega + \text{BI} \tag{6.44}$$

where BI is the boundary integral term, details of which are not important for our discussion.

Here it is important to note that the stabilization integral is directly added to the GFEM weak form and no integration by parts is applied to the high order derivatives in it.

Combining the terms with the unknown T in one integral and putting the remaining terms to the right hand side we get

$$\int_{\Omega^e}\left[U\left(w + \frac{\beta h^e}{2}\frac{dw}{dx}\right)\frac{dT}{dx} + \nu\frac{dw}{dx}\frac{dT}{dx} + \nu\frac{\beta h^e}{2}\frac{dw}{dx}\frac{d^2T}{dx^2}\right]d\Omega = \int_{\Omega^e} f\left(w + \frac{\beta h^e}{2}\frac{dw}{dx}\right)d\Omega + \text{BI} \tag{6.45}$$

Using $T = \sum_{j=1}^{NEN} T_j\,S_j$ and $w = S_i$ we get the following elemental stiffness matrix and force vector

$$K^e = \int_{\Omega^e}\left[U\left(S_i + \frac{\beta h^e}{2}\frac{dS_i}{dx}\right)\frac{dS_j}{dx} + \nu\frac{dS_i}{dx}\frac{dS_j}{dx} + \nu\frac{\beta h^e}{2}\frac{dS_i}{dx}\frac{d^2S_j}{dx^2}\right]d\Omega \tag{6.46}$$

$$F^e = \int_{\Omega^e} f\left(S_i + \frac{\beta h^e}{2}\frac{dS_i}{dx}\right)d\Omega \tag{6.47}$$

As mentioned previously SUPG, being a consistent Petrov GFEM, modifies both K^e and F^e of GFEM. If we use linear elements, i.e. linear shape functions, the term with the second derivative of the unknown becomes zero and the elemental stiffness matrix reduces to

$$K^e = \int_{\Omega^e}\left[U\left(S_i + \frac{\beta h^e}{2}\frac{dS_i}{dx}\right)\frac{dS_j}{dx} + \nu\frac{dS_i}{dx}\frac{dS_j}{dx}\right]d\Omega \tag{6.48}$$

which is the same as the one obtained in Section 6.7 using an inconsistent SU approach. But note that this similarity is valid only for the use of linear elements. Even if we use linear elements, SUPG formulation will provide better results compared to inconsistent Petrov Galerkin due to the differences in the calculation of F^e.

6.9.2 Galerkin Least Squares (GLS) Stabilization

An alternative stabilization is known as Galerkin Least Squares (GLS) formulation for which the $\mathcal{P}(\ )$ operator is selected to be the same as the $\mathcal{L}(\ )$ operator

$$\text{GLS:} \qquad \mathcal{P}(w) = \mathcal{L}(w) = \vec{V} \cdot \nabla w - \nabla \cdot (\nu \nabla w) \tag{6.49}$$

As seen the first term of $\mathcal{P}$ operator is the same as the one used for SUPG. The new second term has second order derivatives and it vanishes for linear elements. Therefore for AD equation solved with linear elements SUPG and GLS provides the same result. For higher order elements or for different DEs (like advection-diffusion-reaction equation) this will not be the case.

6.10 Comparison of Different Formulations – 1D

Consider the following 1D AD problem

$$U\frac{dT}{dx} - \nu\frac{d^2T}{dx^2} = 10e^{-5x} - 4e^{-x} \qquad \text{for} \quad x \in [0,1]$$

$$u(0) = 0 \quad , \quad u(1) = 1$$

A non-constant force function is selected on purpose so that the difference between different formulations can be seen clearly. The problem is solved on a mesh of 10 linear, equi-sized elements using two different Peclet numbers, 0.25 and 5 by taking $U = 1$ and changing ν. Figure 6.5 compares the performance of Galerkin, inconsistent SU and SUPG formulations.

Although the optimum value of $\beta = \coth(Pe) - 1/Pe$ is used for inconsistent SU and SUPG, exact solution at the nodes cannot be obtained due to the force function being not constant. Note that optimum value of β was derived back in Section 6.4 for the constant f case. As seen from Figure 6.5 all three formulations provide good results for low Pe case. However, stabilization is necessary (for the used 10 element mesh) to get acceptable results for high Pe case. It is also clear that SUPG performs better than inconsistent SU. Although not presented, as mentioned previously GLS stabilization would give the same result as SUPG for this problem, when used with linear elements.

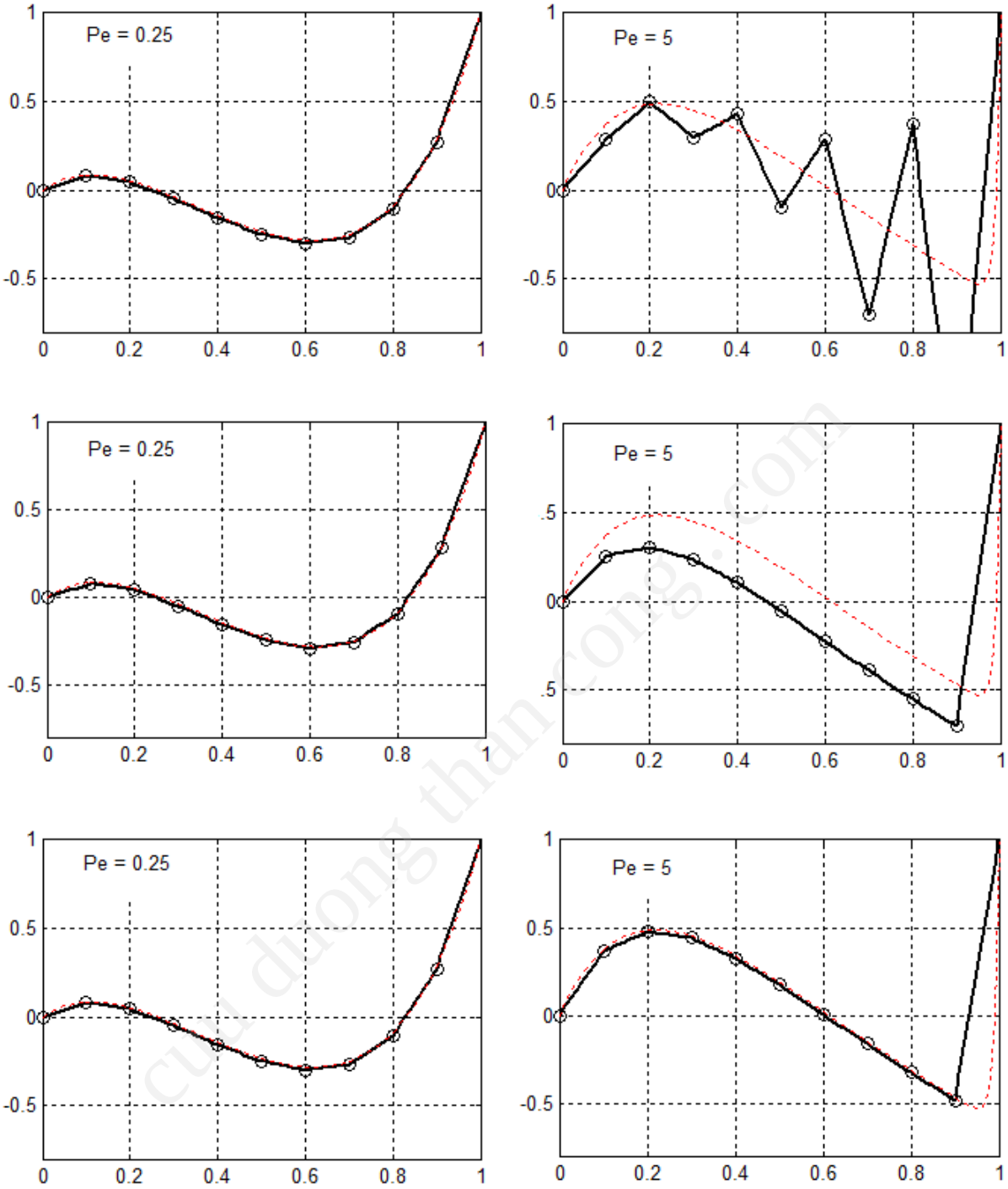


Figure 6.5 Comparison of different stabilizations for 1D AD equation with non-constant force function using 10 equi-sized elements and optimum β value. Top: GFEM, Center: Inconsistent SU, Bottom: SUPG

6.11 Comparison of Different Formulations – 2D

This is a classical test problem used to study two-dimensional AD equation. Problem domain is a unit square, as shown in Figure 6.6, discretized using 10x10=100 equi-sized quadrilateral elements. The velocity field has a constant direction with a constant magnitude of $\|\vec{V}\| = 1$. Diffusivity is taken as $\nu = 10^{-4}$ making the problem convection dominated. Left and top boundaries are inflow boundaries with specified EBCs. Right and bottom boundaries are outflow boundaries. It is possible to use either zero NBCs or zero EBCs at the outflow boundaries. The former is used in the current solutions. Exact solution is the convection of the discontinuity at the left boundary (at point (0,0.8)) into the domain in the direction of the specified velocity. However, this BC jump makes the problem a difficult one for GFEM, which does not use any artificial dissipation.

Figure 6.7 shows the results obtained using Galerkin and SUPG formulations. As seen SUPG provide results that are more diffusive and less oscillatory compared to GFEM.

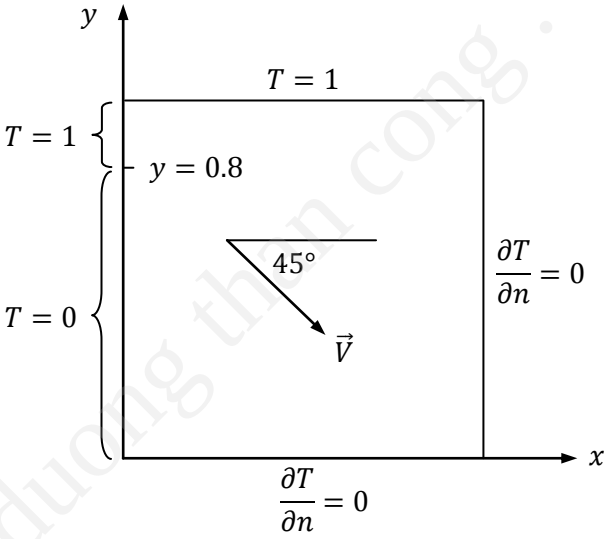


Figure 6.6 Definition of Problem 2, known in the literature as “advection skew to the mesh”

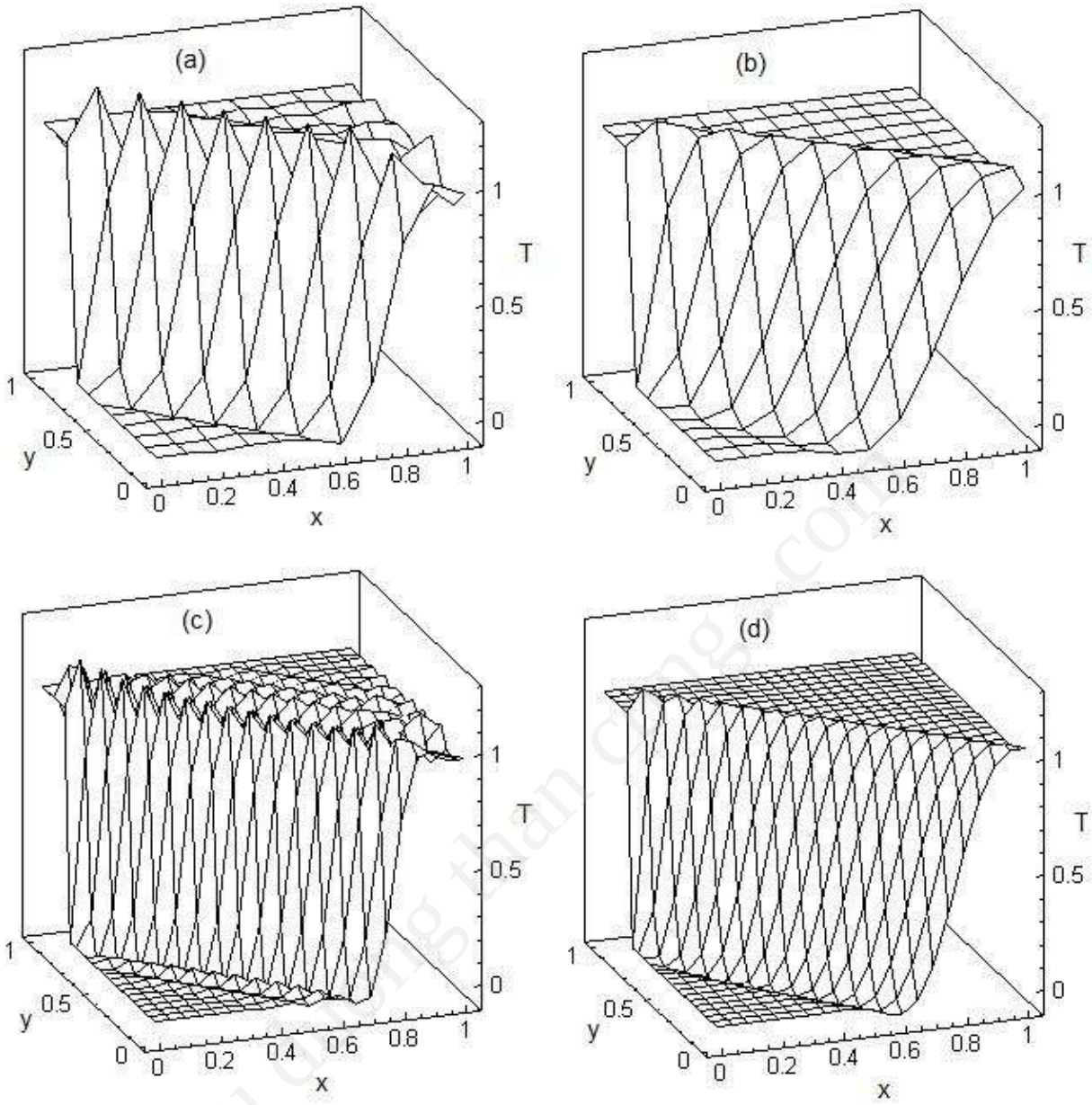


Figure 6.7 Results of Problem 2 using (a) GFEM, NE=100, (b) SUPG, NE=100, (c) GFEM, NE=400, (d) SUPG, NE=400

6.12 Relation Between Oscillatory Solutions and Mesh Quality – Adaptive Mesh Refinement in 1D

Oscillatory behavior of GFEM for convection dominated problems is closely related to the mesh used for the solution. As a demonstration, Figure 6.8 compares the solution of 1D AD equation with $U = 1,\ \nu = 0.01$ and $f = 1$ using GFEM with 10, 20 and 30 equi-sized elements. Given parameters correspond to elemental Peclet numbers of 5, 2.5 and 1.67 for meshes of 10, 20 and 30 equi-sized elements, respectively. As expected, refined meshes provide better results. The problem with using large elements is that these elements can not properly capture the boundary layer (high gradient region) developing at the right boundary. A fourth solution is also obtained for a mesh of 10

elements clustered to the right end of the domain. As seen from the results clustered mesh provides the best approximation, without any oscillations. Note that these are all GFEM results without any upwinding or artificial diffusion.

It is clear that enough effort should be put to the generation of a high quality mesh for accurate and efficient numerical solution of a problem. However, for 2D and 3D problems with more complicated DEs it is not this easy to automatically generate meshes with fine enough elements at necessary regions to capture the physics correctly. In such cases, it is possible to solve the problem with an initial mesh and estimate the error in the approximate result (a posteriori error estimation) and perform solution based adaptive mesh refinement (AMR) to modify the mesh in necessary regions and solve the problem again. A simple AMR application in 1D is provided in the rest of this section.

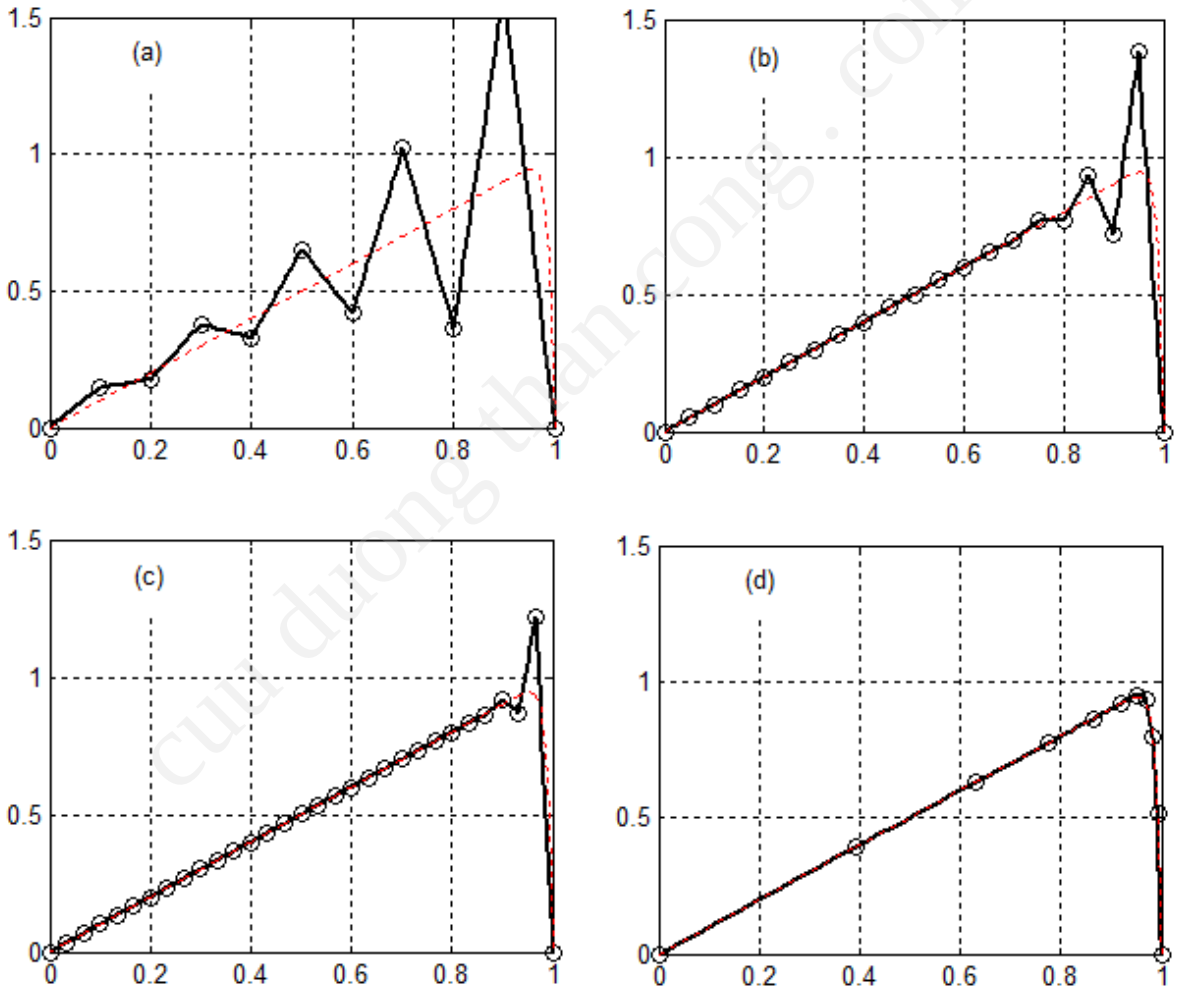


Figure 6.8 Comparison of GFEM results obtained with different meshes for 1D AD equation for $(U = 1,\ \nu = 0.01)$ and $f = 1$. (a) 10 equi-sized elements, (b) 20 equi-sized elements, (c) 30 equi-sized elements, (d) 10 elements clustered to the right boundary.

When linear elements are used for 1D problems, a piecewise linear solution is obtained. First derivative of the unknown is constant over each element and the second derivative is zero. An estimate of the second derivative at the nodes is known to serve as a good error indicator that can be used for AMR purposes. Consider the inner node n of the 1D mesh shown in Figure 6.9 We want to get an estimate for the second derivative at this node. It is surrounded by elements *e* and *e*+1. Points A and B are the midpoints of these elements.

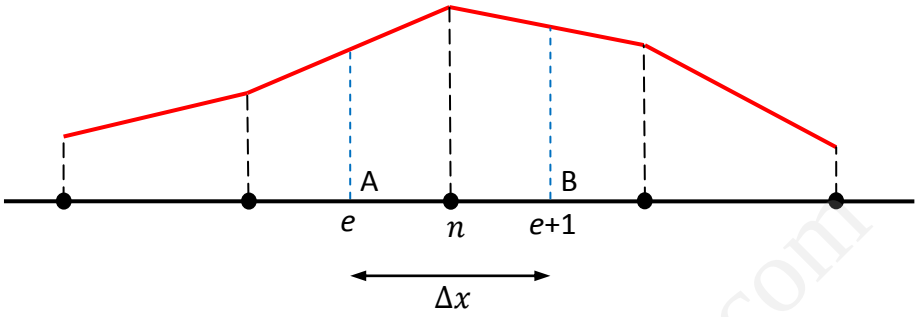


Figure 6.9 Nomenclature used for 1D AMR based on second derivative estimate

The required second derivative can be thought as the derivative of the first derivative as given below

$$\left.\frac{d^2T}{dx^2}\right|_n \approx \frac{\left.\frac{dT}{dx}\right|_B - \left.\frac{dT}{dx}\right|_A}{\Delta x} = \frac{\left.\frac{dT}{dx}\right|_{e+1} - \left.\frac{dT}{dx}\right|_e}{0.5\,(h^{e+1} + h^e)} \tag{6.50}$$

For the nodes at the two ends of the domain, one sided approximations can be used for the second derivative. After estimating the second derivative at each node of the domain, elements that have nodes with second derivatives exceeding a user defined limiting value can be refined by diving them into two.

Figure 6.10 shows the results of this idea applied to the GFEM solution of AD problem studied previously in this section. The solution is started with 10 linear elements and a limiting second derivative value of 20 is used to determine the elements that need to be refined. After the initial solution 8 elements are selected to be refined and the resulting mesh has 18 elements. After the second refinement 6 more elements close to the right boundary are refined, resulting in a mesh of 24 elements. Finally after the third refinement, a mesh of 28 elements is obtained, which provides a solution without oscillations.

In this AMR solution only mesh refinement is implemented. To increase the efficiency of the solution, mesh coarsening can also be applied. For example after the first refinement, solution close to the left boundary is almost linear and very accurate. It is possible to reduce the number of elements in this region by combining two neighboring elements into a single element. Although AMR makes the code intelligent and powerful, it also brings complications to coding.

Keeping element order constant and increasing/decreasing the number of elements, as done in the sample solution explained above, is known as h-refinement. It is also possible to do p-refinement, in which element number is kept the same but order of the elements are increased/decreased as necessary. When these two strategies are used together it is known as hp-refinement. A final refinement alternative, known as r-refinement keeps both element number and their order the same

but relocates the nodes and elements of the mesh so that finer mesh is obtained in high gradient regions.

AMR can also be used for 2D problems, however, dividing large elements into smaller elements or merging small elements into larger ones is not as easy as 1D.

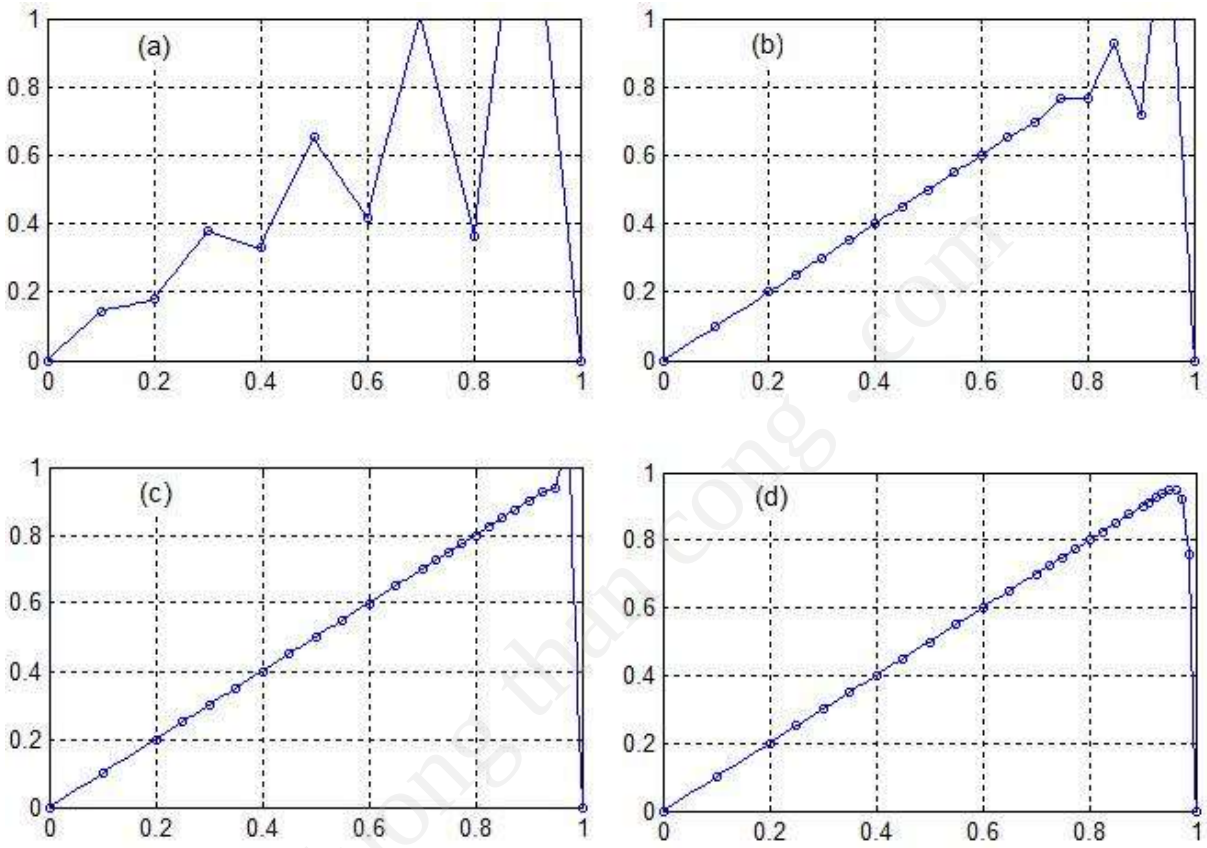


Figure 6.10 Adaptive Mesh Refinement (AMR) applied to GFEM solution of 1D AD equation. (a) Initial mesh with NE=1, (b) 1st refinement, NE=18, (c) 2nd refinement, NE=24, (d) 3rd refinement, NE=28

6.13 Time Dependent Pure Advection in 1D

Consider the following unsteady, 1D pure advection equation

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} = 0 \qquad \text{for} \quad x \in [0, 1.4] \;\; , \;\; t \in [0, 1]$$

$$T(0,t) = 0$$

$$T(x,0) = \begin{cases} \text{if } x < 0.4 & \dfrac{1 + \cos\left(5\pi(x-0.2)\right)}{2} \\ \text{else} & 0 \end{cases}$$

Initial condition given at $t = 0$ is a smooth cosine hill with height of 1 centered at point $x = 0.2$. This hill will travel in the positive x direction with a velocity of $U = 1$. Since this is a pure advection problem, the shape of the hill is expected to be the same as it travels. The equation is hyperbolic in nature and only one BC at the inflow (left end of the domain) is specified. In the 1D unsteady code zero NBC can be provided at the right boundary since it corresponds to a "do nothing" BC.

Forward differencing (FD), backward differencing (BD) and Crank-Nicolson (CN) with 100 linear elements are used for the solution. Figure 6.11 provides the results at $t = 1$ for FE scheme used with different time steps. Being an explicit scheme FD has time step restrictions to get stable solutions. Only for small enough time steps the solution is satisfactory. When time step is increased, unphysical waves develop behind the main traveling hill and they grow in magnitude. These waves are an indication of dispersion errors of the approximation.

Figure 6.12 provides the results for BD scheme. This scheme is implicit and it is unconditionally stable for any time step value. As seen even for very large time steps the solution is free of oscillations, i.e. it is bounded. However, for large time steps the solution is too diffusive, as seen by the drop of the hill's height. This drop is an indication of diffusion errors.

Finally Figure 6.13 provides the results for CN scheme. Unlike the first two, which are first order accurate in time, CN is second order accurate. For small time steps its diffusive error is less than BD scheme. However it is not as "oscillation-free" as the BD scheme with large time steps.

Note that the cosine hill travels a distance of 1 unit with a unit velocity, i.e. total integration time is also 1 unit. For a time step value of 0.1 the hill travels a distance of 0.1 units. For a mesh of 100 elements, as used in all these simulations, one element length is 0.014 units. Therefore with Δt=0.1, the hill try to pass over about 7 elements in one time step, which is just too much for certain schemes.

The above calculation is usually presented in terms of the nondimensional Courant number defined as

$$C = \frac{U\Delta t}{h^e} \tag{6.51}$$

Courant number is important in studying stability characteristics of time integration schemes. For $U = 1, \Delta t = 0.1$ and $h^e = 0.014$ Courant number turns out to be $C = 7.14$, which exceeds the Courant Friedrichs Lewy (CFL) stability criteria of certain schemes. CFL criterion provides the maximum time step that can be used to get stable solutions for a given element length. If we decrease the element length to get more accurate solutions, we should also decrease the time step so that the CFL criterion is not violated.

We have to be careful when deriving conclusions based on the solutions of a single problem. For example this problem is purely advective, i.e. there is no physical diffusion. Real life problems usually include some amount of physical diffusion which will stabilize the solutions to a certain degree. Shape of the initial cosine hill is also important. For example if we try to work with a square wave with sharp corners, the solutions will be much more oscillatory.

Although not presented here, it is possible to use stabilized formulations such as SUPG or GLS for time dependent problems. Taylor-Galerkin Method is another popular stabilization technique used to solve unsteady problems. You can find its details in reference [1] and in several articles.

Adaptive Mesh Refinement (AMR) discussed in the previous section can be used for time dependent problems, too. Actually since the solution and its gradients are changing in time, solution based mesh adaptation becomes a necessity for designing efficient solvers for unsteady problems.

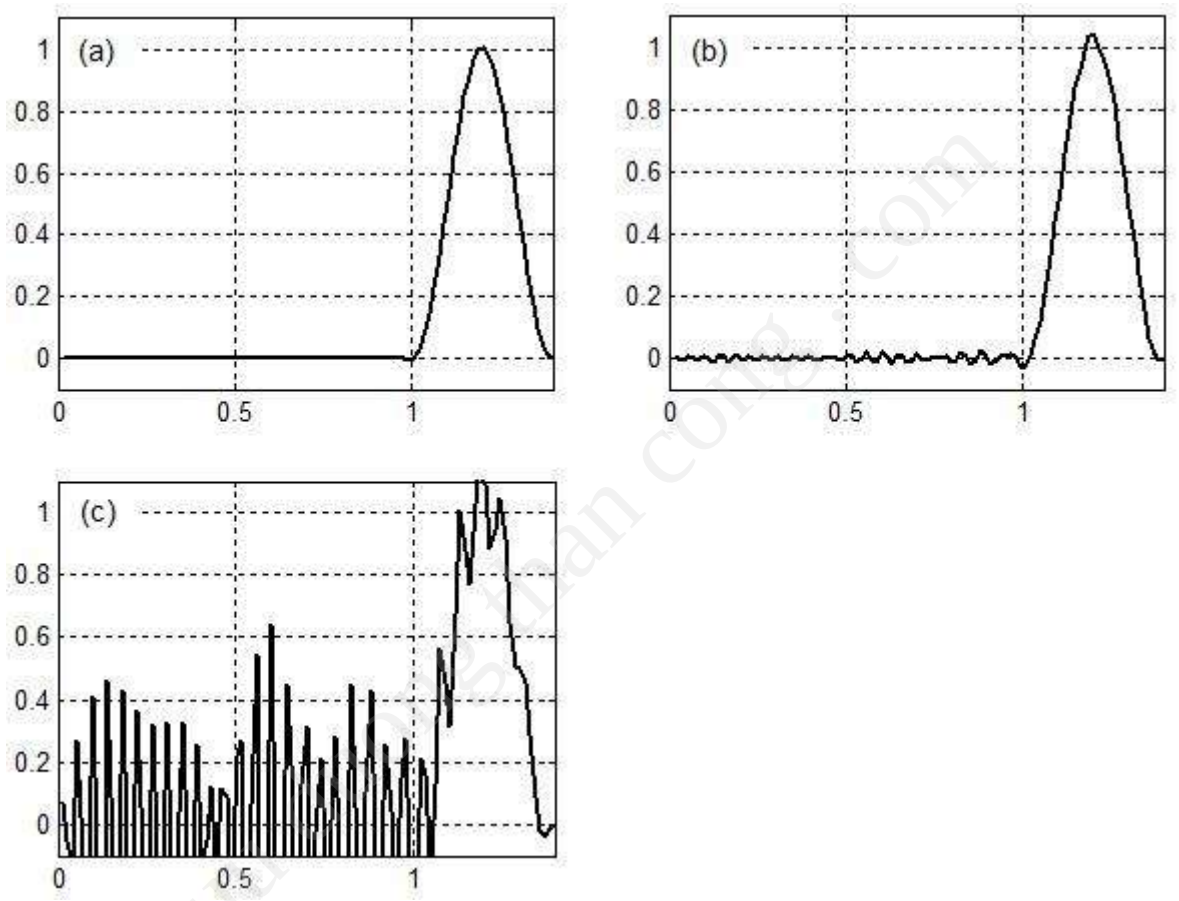


Figure 6.11 Solution of Problem 1 at t = 1 using FD scheme with different time steps.
(a) $\Delta t = 0.0001$, (b) $\Delta t = 0.005$, (c) $\Delta t = 0.001$

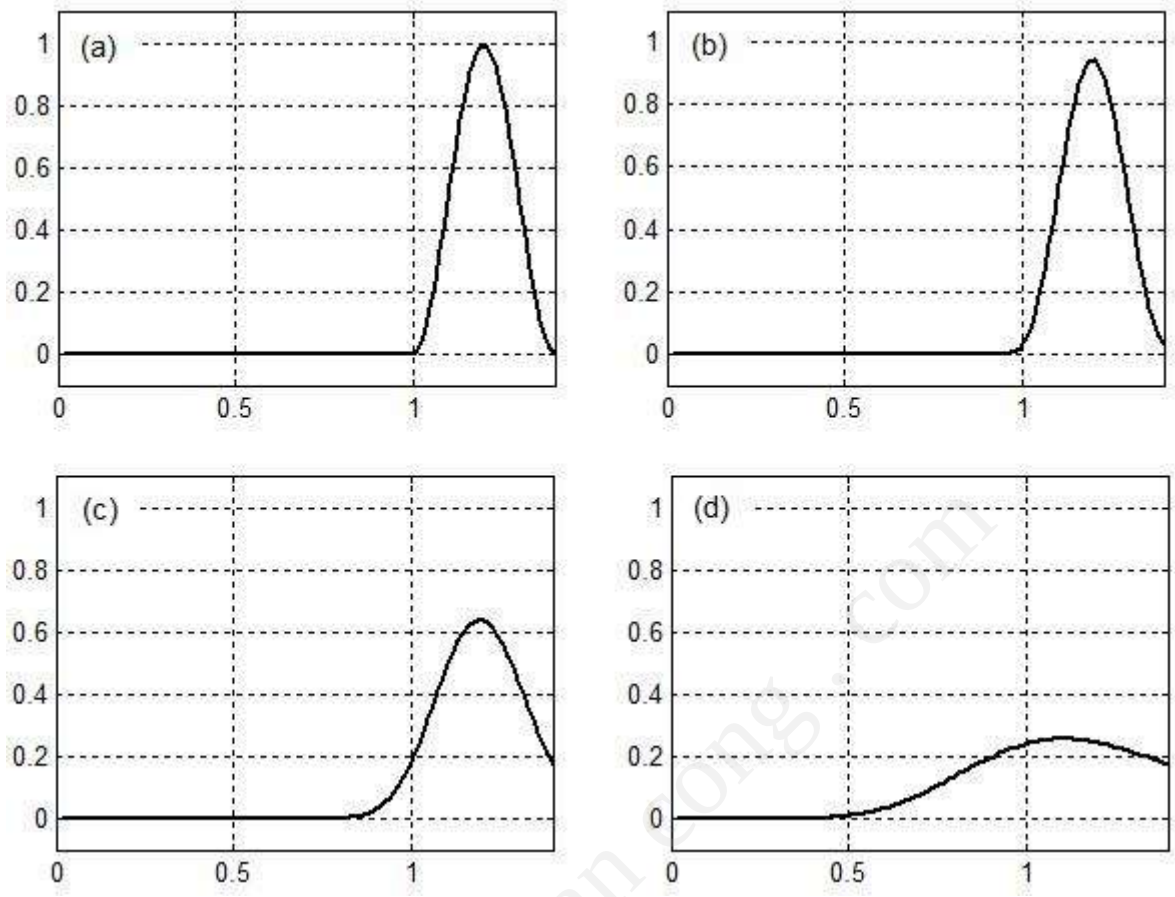


Figure 6.12 Solution of Problem 1 at $t = 1$ using BD scheme with different time steps.
(a) $\Delta t = 0.0001$, (b) $\Delta t = 0.001$, (c) $\Delta t = 0.01$, (d) $\Delta t = 0.1$

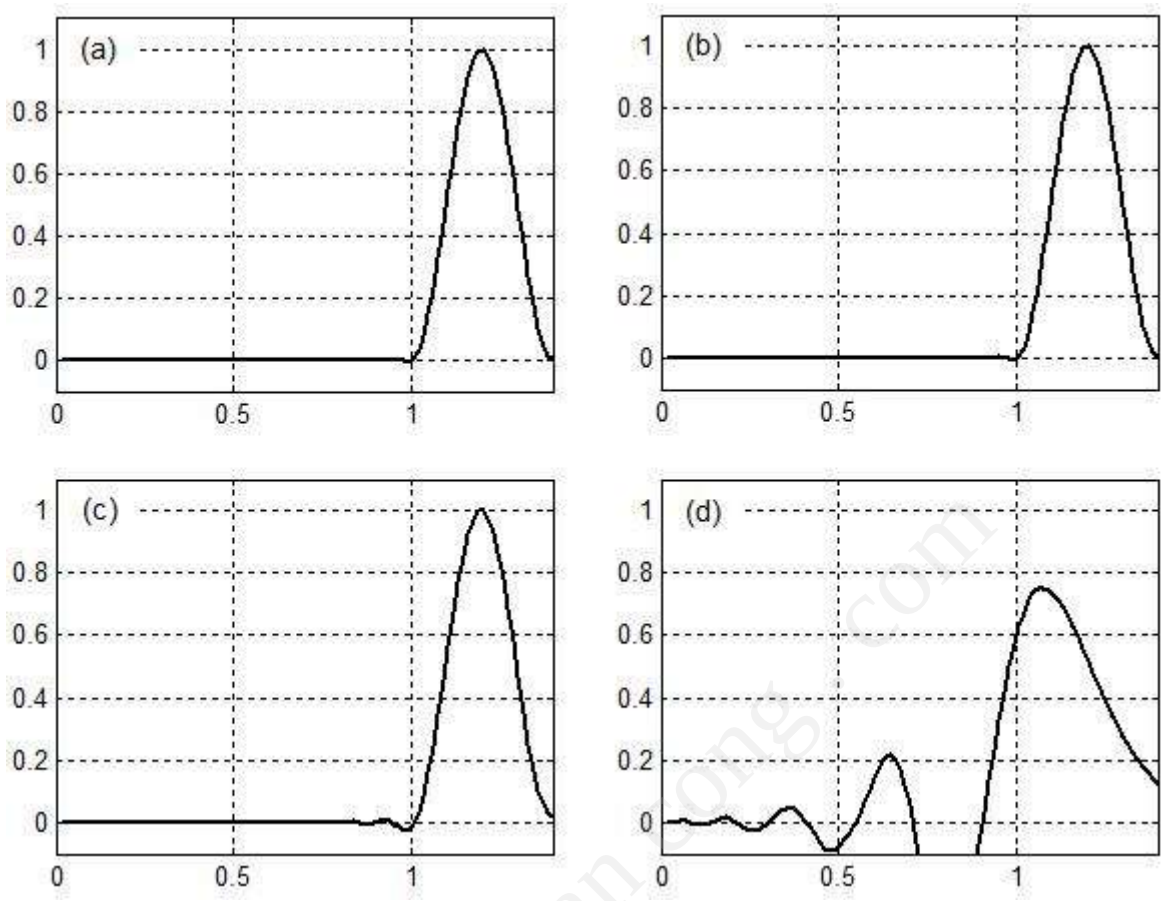


Figure 6.13 Solution at t = 1 using CN scheme with different time steps.
(a) Δt = 0.0001, (b) Δt = 0.001, (c) Δt = 0.01, (d) Δt = 0.1

6.14 Time Dependent Pure Advection in 2D

2D version of the previous problem, known in the literature as "rotating cosine hill" is sketched in Figure 6.14. Governing DE and supporting boundary and initial conditions are

$$\frac{\partial T}{\partial t} + \vec{V} \cdot \nabla T = 0 \ , \qquad \Omega \in [-1,1] \times [-1,1] \ , \qquad t \in [0, 6\pi]$$

$$T = 0 \qquad \text{at all boundaries}$$

$$T(x,y,0) = \begin{cases} \frac{1}{2}[\cos(4\pi r) + 1] & \text{if} \quad r < 0.25 \\ 0 & \text{else} \end{cases} \qquad \text{where} \qquad r = \sqrt{(x+0.5)^2 + y^2}$$

Problem domain is a 2x2 square. Initially the unknown scalar T is specified to be zero everywhere except the cosine hill centered at point A with a height of 1 and a diameter of 0.5. A velocity field is specified as $\vec{V} = -y\vec{\imath} + x\vec{\jmath}$, which is used to rotate the cosine hill around the origin in CCW direction. EBC of value zero is given on all boundaries. Similar to the previous problem, this is also a pure

advection case and the exact solution should have the hill rotating around the origin, without changing its shape. Unfortunately due to diffusive errors of a numerical solution, the hill will diffuse out as it rotates, which will be seen as a decrease in its height and an increase in its radius. Also due to the dispersive errors, there will be trailing waves behind the rotating hill. The simulations are run for a total time of 2π, which corresponds to one revolution of the hill.

Figures 6.15, 6.16 and 6.17 show the results obtained using 100, 400 and 900 quadrilateral elements. For each case Backward Difference (BD) and Crank-Nicolson (CN) schemes are tested with two different time steps. Considering its unstable charateristics, Forward Difference (FD) scheme is not used for this problem.

As seen from Figure 6.15, a mesh of 100 elements provides totally unacceptable results. Altough not reported here, even smaller time steps are tried without any success. This mesh resolution is simply not enough to capture the time dependent physics of this advection problem.

With 400 elements (Figure 6.16), it is possible to clearly identify the roatting hill. Similar to the previous 1D problem, BD scheme is more diffusive than CN. As expected, using a smaller time step ends up with better results. For 400 elements with CN scheme and a time step value of $\pi/1000$, height of the hill after one rotation reduces to 0.75 (it should be 1 for the exact solution) and trailing waves have a maximum undershoot of 0.17.

With 900 elements (Figure 6.17), BD is still too diffusive for the larger time step and the result is unacceptable. When the time step is reduced BD provides acceptable results. Best reult is obtained using CN with the smaller time step, for which height of the hill after one rotation reduces to 0.92 and trailing waves have a maximum undershoot of 0.08.

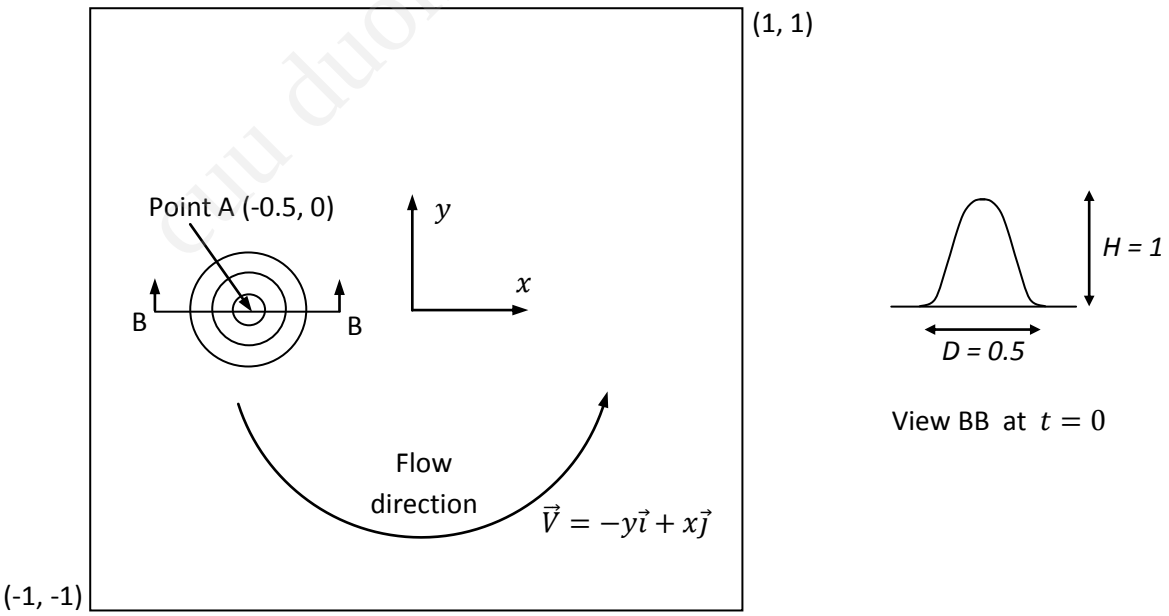


Figure 6.14 Definition of the "Rotating Cosine Hill" problem

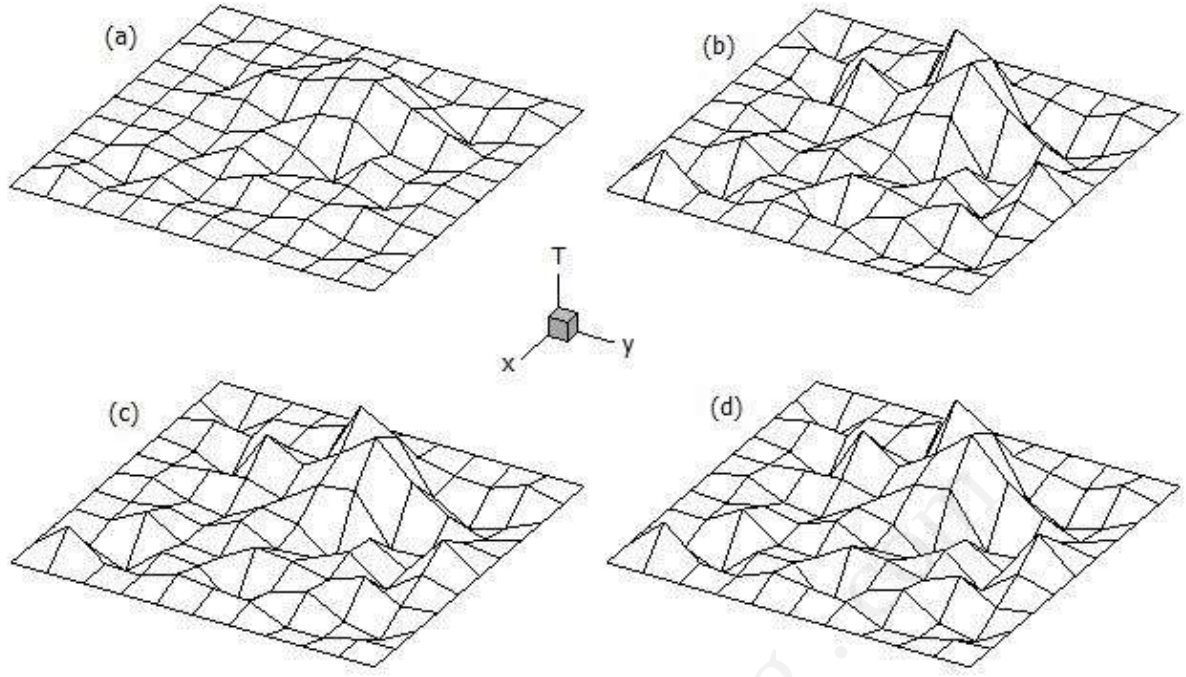


Figure 6.15 Results after one rotation using 100 quadrilateral elements

(a) BD, $\Delta t = \pi/100$, (b) CN, $\Delta t = \pi/100$, (c) BD, $\Delta t = \pi/1000$, (d) CN, $\Delta t = \pi/1000$

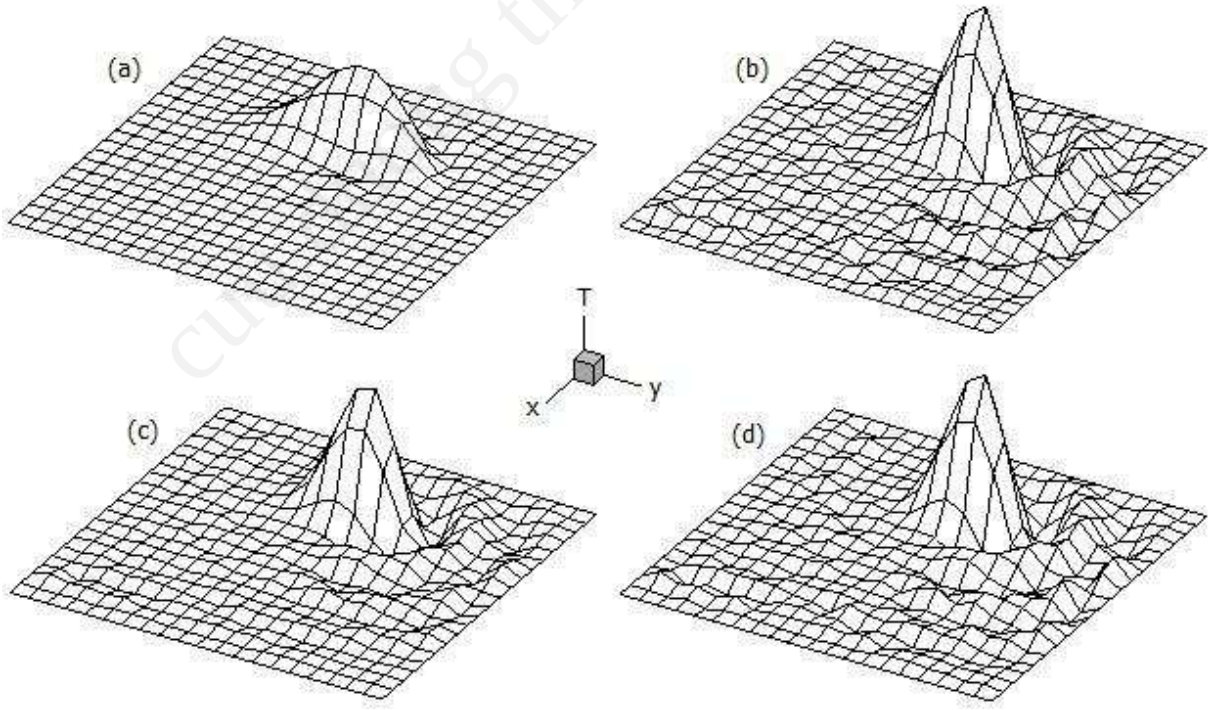


Figure 6.16 Results after one rotation using 400 quadrilateral elements

(a) BD, $\Delta t = \pi/100$, (b) CN, $\Delta t = \pi/100$, (c) BD, $\Delta t = \pi/1000$, (d) CN, $\Delta t = \pi/1000$

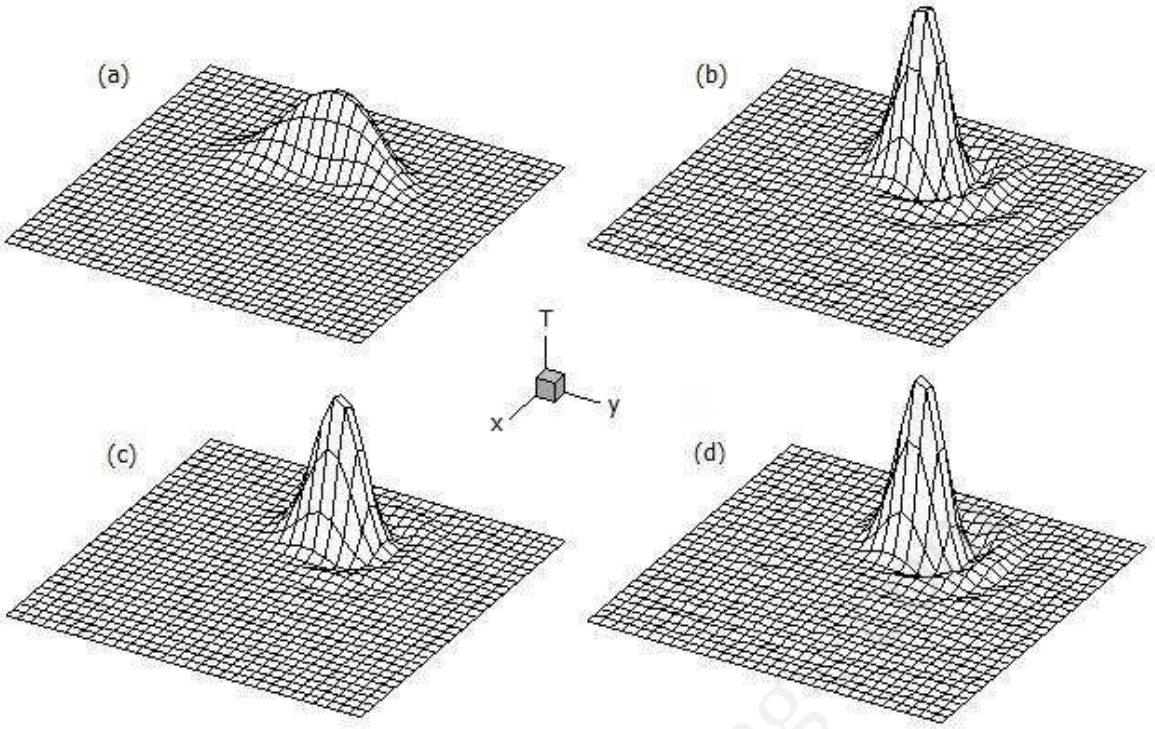


Figure 6.17 Results after one rotation using 900 quadrilateral elements

(a) BD, $\Delta t = \pi/100$, (b) CN, $\Delta t = \pi/100$, (c) BD, $\Delta t = \pi/1000$, (d) CN, $\Delta t = \pi/1000$

Both diffusive and dispersvive errors accumulate as the time elapses and we expect to see worse results if the hill is allowed to make more rotations. For example if the 900 element case with CN scheme and $\Delta t = \pi/1000$ is run for 3 rotations, maximum height of the hill reduces to 0.80 and the trailing waves provide a maximum undershoot of 0.17. Compare these numbers with the ones given in the previous paragraph.

6.15 Exercises

E-6.1. What happens if the modified shape functions of Section 6.7 are not only used in integrating the convective term, but also the diffusive term? Will the discontinuity of the modified shape functions possess any difficulty?

E-6.2: Consider a velocity field purely in the x direction, i.e. $\vec{V} = V_1\vec{\imath} + 0\vec{\jmath}$. For a square quadrilateral element with its edges aligned nicely with x and y axes, show that the artificial diffusivity matrix given in Eqn (6.32) becomes

$$\tilde{\nu} = \bar{\nu}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Note that only one component of the artificial diffusion matrix is nonzero. This is how crosswind diffusion is eliminated in streamline upwinding.

E-6.3: For the velocity field given in the previous exercise determine the artificial diffusion matrix, but this time consider that the square element is rotated CCW by 45^o so that its edges are no longer aligned with the x and y axes.

E-6.4: As mentioned in Section 6.13 GLS and SUPG type stabilization techniques can be used for time dependent problems too. Modify the unsteady 1D code to include these stabilizations and solve the problem of Section 6.13. Compare the stabilized results with the provided ones.

E-6.5: Repeat exercise E-6.4 for 2D problems. Modify 2D unsteady code to include stabilization terms and solve the rotating cosine hill problem of Section 6.13. Compare the stabilized results with the provided ones.

E-6.6: 2D unsteady problems can be costly in terms of computation time. Measure the computation time of rotating cosine hill problem of Section 6.14 on your own computer for different meshes. Use MATLAB's profiler to determine the most time consuming parts of the code. Suggest ideas to reduce run time. Implement your ideas and present the improvements.

6.16 References

[1] J. Donea, A. Huerta, Finite Element Methods for Flow Problems, John Wiley and Sons, 2003.