Chapter 8

Finite Element Methods for Hyperbolic Systems

8.1. Principle for one-dimensional scalar laws

8.1.1. *Weak form*

The conservation form [1.1] of a scalar hyperbolic conservation law (see section 1.1.1) is recalled:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S$$

This equation may also be written as in equation [3.20], recalled here:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} - S = 0$$

Assume that equation [1.1] is to be solved over a computational domain [0, *L*], with initial and boundary conditions defined so as to guarantee solution existence and uniqueness (see section 1.2.2 for details).

The first step consists of multiplying equation [3.20] with a so-called weighting function *w*(*x*, *t*), also called a test function:

Wave Propagation in Fluids: Models and Numerical Techniques, Second Edition Vincent Guinot

$$\left(\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}-S\right)w=0 \qquad \begin{cases}\forall w(x,t)\\ \forall(x,t)\end{cases} \qquad [8.1]$$

Equation [8.1] is integrated over the solution domain [0, L] between times t_1 and t_2:

$$\int_{t_1}^{t_2}\int_0^L\left(\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}-S\right)w\,\mathrm{d}x\;\mathrm{d}t=0 \qquad [8.2]$$

Equation [8.2] is a particular case of the weak form [3.21] seen in section 3.4.1. Equation [8.2] is obtained from equation [3.21] by setting $x_1 = 0$ and $x_2 = L$. Finite element methods seek a solution to equation [8.2].

Note that the non-conservation form [1.17] of the equation (see section 1.1.3 for more details) may also be solved. Reproducing the above reasoning for equation [1.17] yields:

$$\int_{t_1}^{t_2}\int_0^L\left(\frac{\partial U}{\partial t}+\lambda\frac{\partial U}{\partial x}-S'\right)w\,\mathrm{d}x\;\mathrm{d}t=0 \qquad [8.3]$$

As mentioned in Chapter 3, equations [1.1], [1.17], [8.2] and [8.3] are equivalent as long as the solution U is continuous. When the solution is discontinuous, the choice of the formulation and the choice of the test function w may have important consequences on the behavior of the solution (see section 8.5).

Solving the weak forms [8.2] or [8.3] amounts to solving the original conservation law in an average sense over the solution domain. The averaging is a function of the weighting function w used. For this reason, the approach is sometimes referred to as the "weighted residuals" method.

8.1.2. *Discretization of space and time*

8.1.2.1. *Principle*

Finite element methods [HER 07] are similar to finite difference and finite volume approaches in that they solve a discretized version of the governing equations. Space is discretized into pre-defined computational points (called nodes) and time is discretized into pre-defined time levels (Figure 8.1). In contrast with finite differences (see section 6.1.1), finite element methods do not seek the computational solution at the computational nodes only. The finite element solution is defined at all points of the domain [0, L] for a given time. It is sought in the form:

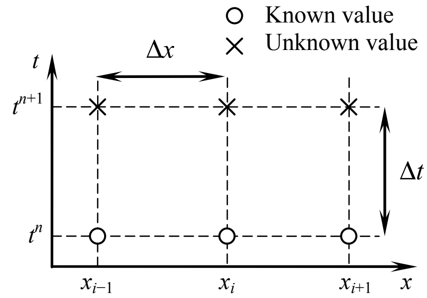

Figure 8.1. *Discretization of space and time for a one-dimensional problem*

$$U(x,t^n) = \sum_{i=1}^{M} U_i^n s_i(x) \tag{8.4}$$

where functions $s_i(x)$ are so-called shape functions. They are defined *a priori*. The U_i^n are coefficients allowing for a linear combination of the shape functions. The solution is known completely at time level n if all the coefficients U_i^n can be computed.

The flux and source term are sought in the form:

$$\left.\begin{aligned} F(x,t^n) &= \sum_{i=1}^{M} F_i^n s_i(x) \\ S(x,t^n) &= \sum_{i=1}^{M} S_i^n s_i(x) \end{aligned}\right\} \tag{8.5}$$

In classical finite element approaches, s_i is equal to 1 at node i and is zero at all other nodes. It is non-zero over the interval $]x_{i-1}, x_{i+1}[$ (Figure 8.2).

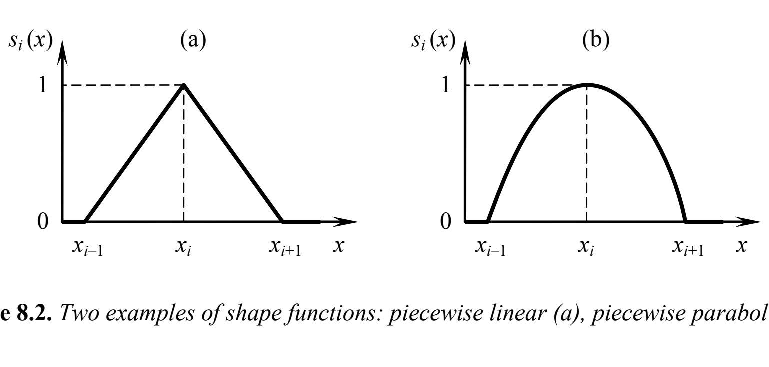

Figure 8.2. *Two examples of shape functions: piecewise linear (a), piecewise parabolic (b)*

8.1.2.2. *Discretization of the conservation form*

Let $t_1 = t^n$ and $t_2 = t^{n+1}$ in equation [8.2]. Swapping the time and space integrals leads to:

$$\int_0^L [U^{n+1}(x) - U^n(x)] w \, \mathrm{d}x + \int_0^L \int_{t^n}^{t^{n+1}} \left(\frac{\partial F}{\partial x} - S \right) w \, \mathrm{d}t \; \mathrm{d}x = 0 \qquad [8.6]$$

The time integral of $(\partial F / \partial x - S) w$ is approximated as:

$$\int_{t^n}^{t^{n+1}} \left(\frac{\partial F}{\partial x} - S \right) w \, \mathrm{d}t \approx \left[(1-\theta) \left(\frac{\partial F}{\partial x} - S \right)^n + \theta \left(\frac{\partial F}{\partial x} - S \right)^{n+1} \right] \Delta t \, w \qquad [8.7]$$

where θ is an implicitation parameter between 0 and 1. Seeking the solution in the form [8.4–5] and substituting equation [8.7] into equation [8.6], we obtain

$$\begin{aligned} &\int_0^L \sum_{i=1}^M (U_i^{n+1} - U_i^n) s_i w \mathrm{d}x + \Delta t \int_0^L \frac{\partial}{\partial x} \sum_{i=1}^M \left[(1-\theta) F_i^n + \theta F_i^{n+1} \right] s_i w \mathrm{d}x \\ &\qquad + \Delta t \int_0^L \sum_{i=1}^M \left[(1-\theta) S_i^n + \theta S_i^{n+1} \right] s_i w \mathrm{d}x = 0 \end{aligned} \qquad [8.8]$$

Swapping the sum operators and partial derivatives yields

$$\begin{aligned} &\sum_{i=1}^M (U_i^{n+1} - U_i^n) \int_0^L s_i w \mathrm{d}x + \Delta t \sum_{i=1}^M \left[(1-\theta) F_i^n + \theta F_i^{n+1} \right] \int_0^L \frac{\partial s_i}{\partial x} w \mathrm{d}x \\ &\qquad + \Delta t \sum_{i=1}^M \left[(1-\theta) S_i^n + \theta S_i^{n+1} \right] \int_0^L s_i w \mathrm{d}x = 0 \end{aligned} \qquad [8.9]$$

The shape functions s_i and the weighting functions w being known, their integrals over the computational domain are known too. Equation [8.9] may be rewritten in the form:

$$\begin{aligned} &\sum_{i=1}^M \left\{ U_i^{n+1} - U_i^n + \left[(1-\theta) S_i^n + \theta S_i^{n+1} \right] \Delta t \right\} C_{i,w} \\ &+ \sum_{i=1}^M \left[(1-\theta) F_i^n + \theta F_i^{n+1} \right] \Delta t \, D_{i,w} = 0 \end{aligned} \qquad [8.10]$$

where coefficients $A_{i,w}$ and $B_{i,w}$ are given by:

$$\left.\begin{aligned} C_{i,w} &= \int_0^L s_i(x)w(x)\mathrm{d}x \\ D_{i,w} &= \int_0^L \frac{\partial s_i(x)}{\partial x} w(x)\mathrm{d}x \end{aligned}\right\} \qquad [8.11]$$

Equation [8.10] involves the unknown nodal values U_i^{n+1} , F_i^{n+1} and S_i^{n+1} at time $n + 1$. Since F and S are known functions of U, equation [8.10] may be rewritten so as to involve the single unknown U_i^{n+1}.

Denoting by M the number of nodes in the computational domain, M equations [8.10] can be written. To do so, it is sufficient to choose M different weighting functions w_j ($j = 1, 2, \ldots, M$) and determine $C_{i,j}$ and $D_{i,j}$ obtained for each w_j ($j = 1, 2, \ldots, M$).

Computation of the coefficients $C_{i,j}$ and $D_{i,j}$ is facilitated if the support of the weighting functions is narrow. A classical choice consists of using weighting functions w_j that are zero except over the interval $]x_{j-1}, x_{j+1}[$ (see section 8.1.3 for typical examples). In such a case, the coefficients $C_{i,j}$ and $D_{i,j}$ are non-zero only for $j = i - 1, j = i$ or $j = i + 1$. Equation [8.10] is rewritten in the form:

$$\begin{aligned} &\sum_{i=1}^{M} C_{i,j}(U_i^{n+1} + \theta\Delta t\, S_i^{n+1}) + D_{i,j}\theta\Delta t\, F_i^{n+1} \\ &= \sum_{i=1}^{M} C_{i,j}\left[U_i^n - (1-\theta)\Delta t\, S_i^n\right] - (1-\theta)\Delta t D_{i,j} F_i^n \end{aligned} \qquad [8.12]$$

where $C_{i,j}$ and $D_{i,j}$ are defined as:

$$\left.\begin{aligned} C_{i,j} &= \int_0^L s_i(x)w_j(x)\,\mathrm{d}x \\ D_{i,j} &= \int_0^L \frac{\partial s_i(x)}{\partial x} w_j(x)\,\mathrm{d}x \end{aligned}\right\} \qquad [8.13]$$

Writing M equations [8.12] for $j = 1, \ldots, M$, leads to an M×M system of algebraic equations that can be solved uniquely for the nodal values U_i^n.

If *F* and *S* are nonlinear functions of *U*, system [8.12] is nonlinear and must be solved using iterative techniques. If *F* and *S* are linear functions of *U*, [8.12] becomes linear and can be rewritten in the form:

$$\mathrm{R}\mathrm{U}^{n+1} = \mathrm{b} \qquad [8.14]$$

where the components of vector U^{n+1} are the nodal unknowns U_i^{n+1} and the *j*th row of vector b is the right-hand side member of equation [8.12]. Matrix R is often referred to as the mass matrix, or rigidity matrix. Its expression is given in section 8.2 in a number of cases.

8.1.2.3. *Discretization of the non-conservation form*

An alternative option consists of discretizing the non-conservation form [8.3] of the governing equation. This leads to a system involving only the nodal unknowns U_i^{n+1}. Substituting $t_1 = t^n$, $t_2 = t^{n+1}$ and $w = w_j$ in equation [8.3], swapping the time and space integrals leads to:

$$\int_0^L [U^{n+1}(x) - U^n(x)] w_j \,\mathrm{d}x + \int_0^L \int_{t^n}^{t^{n+1}} \left(\lambda \frac{\partial U}{\partial x} - S'\right) w_j \,\mathrm{d}t \,\mathrm{d}x = 0 \qquad [8.15]$$

As in section 8.1.2.2, the time integral of $(\lambda\,\partial U/\partial x - S')w_j$ is approximated as:

$$\begin{aligned}\int_{t^n}^{t^{n+1}} \left(\lambda \frac{\partial U}{\partial x} - S'\right) w_j \,\mathrm{d}t \approx \Bigg[&(1-\theta)\left(\lambda^{n+1/2} \frac{\partial U^n}{\partial x} - S^n\right) \\ &+ (1-\theta)\left(\lambda^{n+1/2} \frac{\partial U^{n+1}}{\partial x} - S^{n+1}\right)\Bigg] w_j \Delta t\end{aligned} \qquad [8.16]$$

Linearizing *S* with respect to *U* leads to:

$$S_i^{n+1} \approx S_i^n + \left(\frac{\partial S}{\partial U}\right)_i^{n+1/2} (U_i^{n+1} - U_i^n) \qquad [8.17]$$

Reasoning along the same line as in section 8.1.2.2, substituting equation [8.17] into equation [8.16] yields:

$$\begin{aligned}&\sum_{i=1}^{M} (C_{i,j} + \theta D_{i,j} \lambda_i^{n+1/2} \Delta t) U_i^{n+1} - C_{i,j} \left(\frac{\partial S}{\partial U}\right)_i^{n+1/2} U_i^{n+1} \Delta t \\ &= \sum_{i=1}^{M} \left[C_{i,j} - (1-\theta) D_{i,j} \lambda_i^{n+1/2} \Delta t\right] U_i^n + C_{i,j} \left[S_i^n - \left(\frac{\partial S}{\partial U}\right)_i^{n+1/2} U_i^n\right] \Delta t\end{aligned} \qquad [8.18]$$

where coefficients $C_{i,j}$ and $D_{i,j}$ are defined as in equation [8.13]. The term $(\partial S/\partial U)_i^{n+1/2}$ may be estimated explicitly from the known value at time level n. It may also be computed iteratively from a linear combination of the values at time levels n and $n + 1$.

Although the non-conservation form of the equation may seem easier to discretize because it leads to a linear system, it must be used with care with conservation laws with discontinuous solutions. As shown in section 8.4.2, the estimate of the wave speed $\lambda_i^{n+1/2}$ strongly influences the accuracy of the numerical solution.

8.1.3. *Classical shape and test functions*

8.1.3.1. *Galerkin technique*

In the Galerkin technique, the shape and weighting functions are taken from the same function space. The simplest possible option is to use $w_i = s_i$. In the case of piecewise linear functions (Figure 8.2a), the following expressions are used for s and w:

$$s_i(x) = w_i(x) = \begin{cases} 0 & \text{if } x \le x_{i-1} \\ \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \le x \le x_i \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} & \text{if } x_i \le x \le x_{i+1} \\ 0 & \text{if } x \ge x_{i+1} \end{cases} \qquad [8.19]$$

The derivative $\partial s_i / \partial x$ is given by:

$$\frac{\partial s_i}{\partial x}(x) = \begin{cases} 0 & \text{if } x \le x_{i-1} \\ \dfrac{1}{x_i - x_{i-1}} & \text{if } x_{i-1} \le x \le x_i \\ \dfrac{-1}{x_{i+1} - x_i} & \text{if } x_i \le x \le x_{i+1} \\ 0 & \text{if } x \ge x_{i+1} \end{cases} \qquad [8.20]$$

Substituting equations [8.19–20] into equation [8.13] yields the following expression for $C_{i,j}$:

$$C_{i,j} = \begin{cases} 0 & \text{if } j < i-1 \\ (x_i - x_{i-1})/6 & \text{if } j = i-1 \\ (x_{i+1} - x_{i-1})/3 & \text{if } j = i \\ (x_{i+1} - x_i)/6 & \text{if } j = i+1 \\ 0 & \text{if } j > i+1 \end{cases} \qquad [8.21]$$

while $D_{i,j}$ is given by:

$$D_{i,j} = \begin{cases} 0 & \text{i } j < i-1 \\ 1/2 & \text{if } j = i-1 \\ 0 & \text{if } j = i \\ -1/2 & \text{if } j = i+1 \\ 0 & \text{if } j > i+1 \end{cases} \qquad [8.22]$$

8.1.3.2. *Petrov-Galerkin techniques*

Petrov-Galerkin (PG) techniques use shape and weighting functions taken from distinct function spaces (see Figure 8.3 for examples). This allows the formulation to be upwinded by giving more weight to upstream nodes and less weight to downstream nodes (Figure 8.3a). This approach is similar to that of upwind finite difference schemes (see section 6.3).

An extreme case (Figure 8.3b) is obtained with a test function w_i that is one over the cell immediately upstream of node i and that is zero everywhere else.

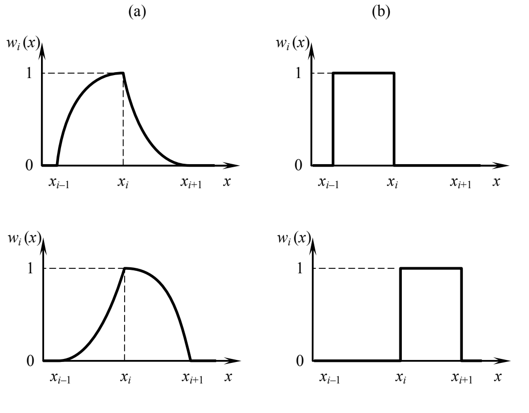

Figure 8.3. *Petrov-Galerkin technique. Typical test functions. General case (a), particular case given by equations [8.23–24] (b). Top: function for a positive wave speed; bottom: functions for a negative wave speed*

In this case, for a positive λ, w_i is defined as follows:

$$w_i(x) = \begin{cases} 0 & \text{for } x \le x_{i-1} \\ 1 & \text{for } x_{i-1} < x \le x_i \\ 0 & \text{for } x > x_i \end{cases} \tag{8.23}$$

while the following definition is obtained for a negative λ:

$$w_i(x) = \begin{cases} 0 & \text{for } x < x_i \\ 1 & \text{for } x_i \le x < x_{i+1} \\ 0 & \text{for } x \le x_{i+1} \end{cases} \tag{8.24}$$

Using equation [8.19] for the shape functions s_i leads to the following expressions for $C_{i,j}$ and $D_{i,j}$ (note that λ is assumed to be positive):

$$\left.\begin{aligned} C_{i,j} &= \begin{cases} 0 & \text{if } j < i \\ (x_i - x_{i-1})/2 & \text{if } j = i \\ (x_{i+1} - x_i)/2 & \text{if } j = i+1 \\ 0 & \text{if } j > i+1 \end{cases} \\ D_{i,j} &= \begin{cases} 0 & \text{if } j < i \\ 1 & \text{if } j = i \\ -1 & \text{if } j = i+1 \\ 0 & \text{if } j > i+1 \end{cases} \end{aligned}\right\} \tag{8.25}$$

8.1.3.3. *A particular case: the SUPG approach*

SUPG stands for Streamline Upwind Petrov-Galerkin. In the SUPG approach, the test function w_i is derived from the shape function s_i as:

$$w_i(x) = s_i(x) + a_i \lambda \frac{\partial s_i}{\partial x} \tag{8.26}$$

where the so-called stabilizing coefficient a_i is a function of the cell size ($a_i > 0$). With equation [8.26], the shape function is distorted by giving more weight to upstream nodes and less to downstream nodes. The consequence is scheme upwinding, with the expected result that solution monotony should be enhanced compared to the original Galerkin technique.

Equation [8.26] is applied to two types of functions hereafter:

1) For triangular shape functions s_i (Figure 8.4a), $\partial s_i / \partial x$ is piecewise constant. It is positive between nodes $i-1$ and i, negative between nodes i and $i+1$. A constant quantity, the sign of which is the same as λ, is added to s_i on the left-hand side of node i. In contrast, a constant quantity that has the sign of λ is subtracted from s_i on the right-hand side of the node. The resulting function w_i (Figure 8.4) is given by:

$$w_i(x) = \begin{cases} 0 & \text{for } x < x_{i-1} \\ \dfrac{x - x_{i-1}}{x_i - x_{i-1}} + \dfrac{a\lambda}{x_i - x_{i-1}} & \text{for } x_{i-1} < x < x_i \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} - \dfrac{a\lambda}{x_{i+1} - x_i} & \text{for } x_i < x < x_{i+1} \\ 0 & \text{for } x < x_{i-1} \end{cases} \qquad [8.27]$$

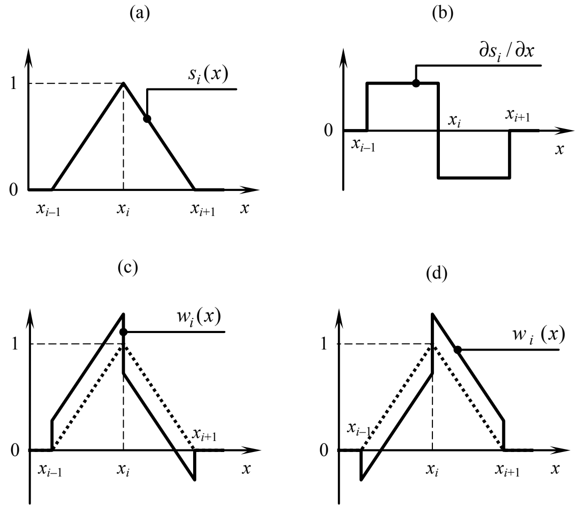

Figure 8.4. *Applying the SUPG approach [8.26] to a triangular shape function. (a) shape function, (b) space derivative of the shape function, (c) SUPG test function for a positive λ, (d) SUPG test function for a negative λ*

2) For parabolic shape functions s_i (Figure 8.4b), $\partial w_i / \partial x$ is linear and decreases linearly between nodes $i-1$ and $i+1$. It is positive at node $i-1$, negative at node $i+1$. The resulting SUPG test function is shown in Figure 8.5.

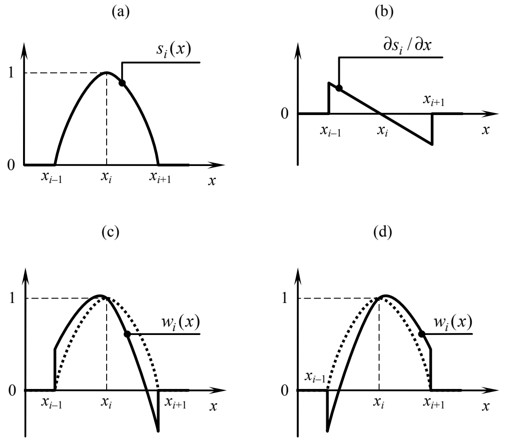

Figure 8.5. *Applying the SUPG approach [8.26] to a parabolic shape function: (a) shape function, (b) space derivative of the shape function, (c) SUPG test function for a positive λ, (d) SUPG test function for a negative λ*

It is easy to check that in the case of triangular shape functions, coefficients $C_{i,j}$ in equation [8.12] are given by:

$$C_{i,j} = \begin{cases} \dfrac{x_i - x_{i-1}}{6} - \dfrac{a\lambda}{2} & \text{if } j = i-1 \\ \dfrac{x_{i+1} - x_{i-1}}{6} & \text{if } j = i \\ \dfrac{x_{i+1} - x_i}{6} + \dfrac{a\lambda}{2} & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases} \qquad [8.28]$$

while coefficients $D_{i,j}$ are given by:

$$D_{i,j} = \begin{cases} \dfrac{1}{2} - \dfrac{a\lambda}{x_i - x_{i-1}} & \text{if } j = i-1 \\ \dfrac{a\lambda}{x_i - x_{i-1}} + \dfrac{a\lambda}{x_{i+1} - x_i} & \text{if } j = i \\ -\dfrac{1}{2} - \dfrac{a\lambda}{x_{i+1} - x_i} & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases} \qquad [8.29]$$

8.2. One-dimensional hyperbolic systems

8.2.1. *Weak formulation*

The conservation form [2.2] of hyperbolic systems of conservation laws is recalled:

$$\frac{\partial \mathrm{U}}{\partial t}+\frac{\partial \mathrm{F}}{\partial x}=\mathrm{S}$$

The weak form of equation [2.2] is:

$$\int_{t_1}^{t_2}\int_{0}^{L}\left(\frac{\partial \mathrm{U}}{\partial t}+\frac{\partial \mathrm{F}}{\partial x}-\mathrm{S}\right)w\,\mathrm{d}x\,\mathrm{d}t=0 \qquad [8.30]$$

The non-conservation form [2.5] is also recalled:

$$\frac{\partial \mathrm{U}}{\partial t}+\mathrm{A}\frac{\partial \mathrm{U}}{\partial x}=\mathrm{S}'$$

where A is the Jacobian matrix of F with respect to U. The weak form of equation [2.5] is:

$$\int_{t_1}^{t_2}\int_{0}^{L}\left(\frac{\partial \mathrm{U}}{\partial t}+\mathrm{A}\frac{\partial \mathrm{U}}{\partial x}-\mathrm{S}'\right)w\,\mathrm{d}x\,\mathrm{d}t=0 \qquad [8.31]$$

Note that the characteristic form [2.24] may also be used:

$$\frac{\partial W_p}{\partial t}+\lambda^{(p)}\frac{\partial W_p}{\partial x}=S_p'', \quad p=1,\ldots,m$$

where W_p is the pth Riemann invariant. The weak form of equation [2.24] is:

$$\int_{t_1}^{t_2}\int_{0}^{L}\left(\frac{\partial W_p}{\partial t}+\lambda^{(p)}\frac{\partial W_p}{\partial x}-S_p''\right)w\,\mathrm{d}x\,\mathrm{d}t=0 \qquad [8.32]$$

Each of these forms has advantages and drawbacks:

– The conservation form is more satisfactory from a theoretical point of view because it remains valid for discontinuous solutions. However, discretizing equation [8.30] for a nonlinear flux function F leads to a nonlinear system that must be solved for the (unknown) nodal values U_i^{n+1} (see section 8.1.2.2). To do so, iterative system inversion techniques must be used, with an increased computational effort.

– The non-conservation form leads to a system in nodal values U_i^{n+1}. However, its performance is highly sensitive to the estimate of the Jacobian matrix A when the solution is discontinuous (see section 8.4.2 for an illustration on a scalar case).

– The characteristic form [8.32] is easy to program because the Riemann invariants can be tracked independently of each other (see section 8.1). Moreover, upwind techniques such as the SUPG approach are easily programmed. However, the characteristic form is not adapted to problems involving shocks. In contrast, it is well-suited to the solution of linear conservation laws or hyperbolic systems such as the water hammer equations or the linear advection equation (see section 8.5.1).

8.2.2. *Application to the non-conservation form*

8.2.2.1. *Galerkin technique*

Reasoning as in section 8.1.2.3, equation [8.31] is transformed into:

$$\int_0^L \left[\mathrm{U}^{n+1}(x) - \mathrm{U}^n(x)\right] w_j \,\mathrm{d}x + \int_0^L \int_{t^n}^{t^{n+1}} \left(\mathrm{A}\frac{\partial \mathrm{U}}{\partial x} - \mathrm{S}'\right) w_j \,\mathrm{d}t\,\mathrm{d}x = 0 \qquad [8.33]$$

The time integral of $(\mathrm{A}\,\partial\mathrm{U}/\partial x - \mathrm{S}')w_j$ is approximated as:

$$\begin{aligned}\int_{t^n}^{t^{n+1}} \left(\mathrm{A}\frac{\partial \mathrm{U}}{\partial x} - \mathrm{S}'\right) w_j \,\mathrm{d}t \approx \Bigg[(1-\theta)\left(\mathrm{A}^{n+1/2}\frac{\partial \mathrm{U}^n}{\partial x} - \mathrm{S}'^n\right) \\ + \theta\left(\mathrm{A}^{n+1/2}\frac{\partial \mathrm{U}^{n+1}}{\partial x} - \mathrm{S}'^{n+1}\right)\Bigg]\Delta t\, w_j\end{aligned} \qquad [8.34]$$

Reasoning along the same line as in section 8.1.2.2, equation [8.33] is transformed into:

$$\begin{aligned}&\sum_{i=1}^{M}(C_{i,j} + \theta D_{i,j}\mathrm{A}_i^{n+1/2}\Delta t)\,\mathrm{U}_i^{n+1} - C_{i,j}\left(\frac{\partial \mathrm{S}'}{\partial \mathrm{U}}\right)_i^{n+1/2}\mathrm{U}_i^{n+1}\Delta t \\ &= \sum_{i=1}^{M}\left[C_{i,j} - (1-\theta)D_{i,j}\mathrm{A}_i^{n+1/2}\Delta t\right]\mathrm{U}_i^n + C_{i,j}\left[\mathrm{S}'^n_i - \left(\frac{\partial \mathrm{S}'}{\partial \mathrm{U}}\right)_i^{n+1/2}\mathrm{U}_i^n\right]\Delta t\end{aligned} \qquad [8.35]$$

where coefficients $C_{i,j}$ and $D_{i,j}$ are defined as in equation [8.13]. Note that $\partial \mathrm{S}'/\partial \mathrm{U}$ is the Jacobian matrix of S' with respect to U.

If shocks are present in the solution, the estimate of $\mathrm{A}_i^{n+1/2}$ may exert a significant influence on the quality of the solution.

8.2.2.2. *Petrov-Galerkin technique*

Recall that the Petrov-Galerkin technique (including the SUPG approach) uses shape and test functions taken from distinct function spaces. In general, the test functions are asymmetric, which introduces upwinding. This is why Petrov-Galerkin techniques are often diffusive, thus allowing artificial oscillations near steep fronts to be eliminated from the solution.

However, upwinding can be applied only if a propagation direction can be identified. In the case of hyperbolic systems, however, the direction in which the waves propagate is not unique. Using fractional steps allows the hyperbolic systems to be broken into several terms. For some of them, a single propagation direction can be identified.

The technique is illustrated for the Saint Venant equations in a rectangular, prismatic channel. For the sake of clarity, the channel is considered frictionless and horizontal. The Saint Venant equations are simplified into the so-called one-dimensional shallow water equations:

$$\begin{gathered}\frac{\partial \mathrm{U}}{\partial t}+\frac{\partial \mathrm{F}}{\partial x}=0\\ \mathrm{U}=\begin{bmatrix}h\\ q\end{bmatrix}\quad \mathrm{F}=\begin{bmatrix}q\\ q^2/h+gh^2/2\end{bmatrix}\end{gathered}\qquad [8.36]$$

The flux function F is broken into two parts F_1 and F_2:

$$\mathrm{F}_1=\begin{bmatrix}q\\ q^2/h\end{bmatrix},\qquad \mathrm{F}_2=\begin{bmatrix}0\\ gh^2/2\end{bmatrix}\qquad [8.37]$$

Upwinding can be applied to the discretization of the flux F_1. The time splitting technique [STR 68, GOU 77] is applied as follows:

1) In a first step, the following equation is solved:

$$\frac{\partial \mathrm{U}}{\partial t}+\frac{\partial \mathrm{F}_1}{\partial x}=0\qquad [8.38]$$

Note that this equation can be written in non-conservation form as:

$$\frac{\partial \mathrm{U}}{\partial t} + \mathrm{A}_1 \frac{\partial \mathrm{U}}{\partial x} = 0 \qquad [8.39]$$

where matrix A_1 is given by:

$$\mathrm{A}_1 = \begin{bmatrix} 0 & 1 \\ -u^2 & 2u \end{bmatrix} \qquad [8.40]$$

This matrix has a double eigenvalue $\lambda^{(1)} = \lambda^{(2)} = u$. Consequently, equation [8.38] (that is only a part of the governing equation [8.36]) is characterized by the single propagation speed $\lambda = u$. This wave speed is used in equation [8.26] if a SUPG technique is to be applied.

2) The solution of equation [8.38] (or equation [8.39], depending on which form of the equation is to be solved) is used as an initial condition to solve the following equation over the computational time step:

$$\frac{\partial \mathrm{U}}{\partial t} + \frac{\partial \mathrm{F}_2}{\partial x} = 0 \qquad [8.41]$$

Equation [8.41] may be solved in conservation form as:

$$\frac{\partial \mathrm{U}}{\partial t} + \mathrm{A}_2 \frac{\partial \mathrm{U}}{\partial x} = 0 \qquad [8.42]$$

where matrix A_2 is defined as:

$$\mathrm{A}_1 = \begin{bmatrix} 0 & 0 \\ c^2 & 0 \end{bmatrix} \qquad [8.43]$$

Steps 1) and 2) are repeated sequentially every time step.

This technique has the advantage that upwinding techniques (such as the SUPG technique) can be used in the first step of the time splitting procedure. Moreover, the approach is easily generalized to multidimensional problems.

8.3. Extension to multidimensional problems

8.3.1. *Weak form of the equations*

For the sake of clarity, only scalar two-dimensional problems are dealt with hereafter. The conservation form of two-dimensional scalar laws is given by equation [5.1], recalled here:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=S$$

where F and G are respectively the fluxes in the x and y directions. The non-conservation form [5.2] is also recalled:

$$\frac{\partial U}{\partial t}+\lambda_x\frac{\partial U}{\partial x}+\lambda_y\frac{\partial U}{\partial y}=S'$$

where λ_x, λ_y and S' are given by equation [5.3], also recalled hereafter:

$$\left.\begin{aligned}\lambda_x&=\frac{\partial F}{\partial U}\\ \lambda_y&=\frac{\partial G}{\partial U}\\ S'&=S-\left(\frac{\partial F}{\partial x}\right)_{U=\text{Const}}-\left(\frac{\partial G}{\partial y}\right)_{U=\text{Const}}\end{aligned}\right\}$$

In practice, [5.1] or [5.2] is solved over a finite two-dimensional domain Ω. As in section 8.1, the weak form of the governing equations is solved. The weak form of the conservation form [5.1] is:

$$\int_{t^n}^{t^{n+1}}\int_{\Omega}\left(\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}-S\right)w\,\mathrm{d}\Omega\,\mathrm{d}t \qquad [8.44]$$

while the weak form of the non-conservation form [5.2] is given by:

$$\int_{t^n}^{t^{n+1}} \int_{\Omega} \left(\frac{\partial U}{\partial t} + \lambda_x \frac{\partial F}{\partial x} + \lambda_y \frac{\partial G}{\partial y} - S' \right) w \, \mathrm{d}\Omega \, \mathrm{d}t \qquad [8.45]$$

As in the case of one-dimensional problems, the solution is sought as a linear combination of pre-defined shape functions s_i over the domain Ω. The weighting functions w_i are also defined over the domain Ω. Space discretization aspects are covered in section 8.3.2. Classical shape and weighting functions are described in section 8.3.3.

8.3.2. *Discretization of space*

In finite element techniques, space is discretized into elements formed by the nodes. These elements form an unstructured grid. The elements are classically triangular or quadrangular (Figure 8.6). Figure 8.6 illustrates the meshing of a two-dimensional domain Ω. The sketch on the left-hand side of the figure shows the computational grid obtained using only triangular elements. The right-hand side sketch in the Figure shows the computational mesh obtained by allowing triangular elements to merge into quadrangular elements. The triangular and triangular-quadrangular meshes have respectively 362 and 211 elements.

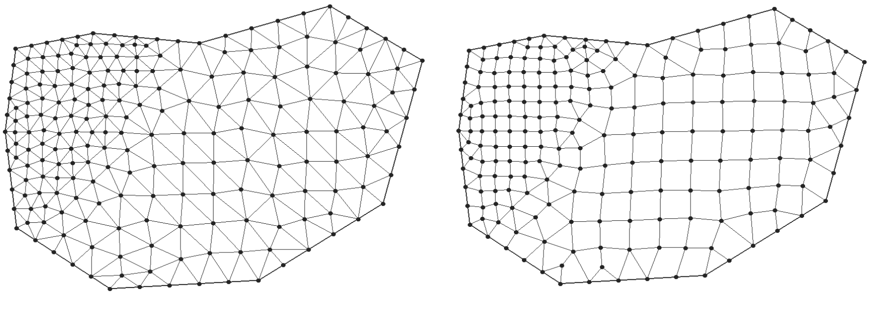

Figure 8.6. *Two examples of finite element meshes for a two-dimensional domain Ω. Left: purely triangular grid (362 elements); right: mixed triangular-quadrangular elements (211 elements)*

8.3.3. *Classical shape and test functions*

The shape functions used for multidimensional finite element methods are classically defined such that s_i is equal to one at the node i and takes a zero value at all other nodes. Figure 8.7 illustrates the example of a piecewise linear function defined over a triangular grid.

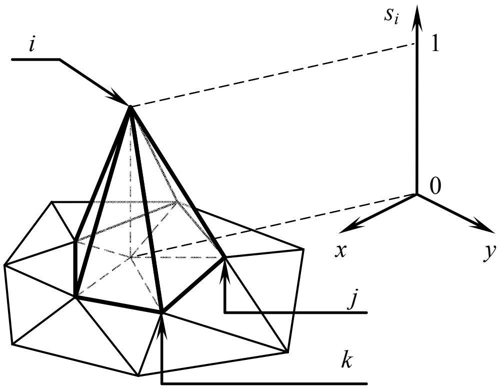

Figure 8.7. *Piecewise linear shape function over a two-dimensional triangular grid. The shape function s_i is equal to 1 at node i and is 0 at all other nodes*

For a triangular element (i, j, k) to which the node i belongs, the shape function s_i is defined as:

$$s_i(x,y) = 1 + E_{i,j,k}(x - x_i) + F_{i,j,k}(y - y_i) \qquad [8.46]$$

where the slopes $E_{i,j,k}$ and $F_{i,j,k}$ are given by:

$$\left.\begin{aligned} E_{i,j,k} &= \frac{x_i - x_j}{(x_j - x_i)^2 + (y_j - y_i)^2} + \frac{x_i - x_k}{(x_k - x_i)^2 + (y_k - y_i)^2} \\ F_{i,j,k} &= \frac{y_i - y_j}{(x_j - x_i)^2 + (y_j - y_i)^2} + \frac{y_i - y_k}{(x_k - x_i)^2 + (y_k - y_i)^2} \end{aligned}\right\} \qquad [8.47]$$

The function s_i is zero over any element of which node i is not a corner node.

Galerkin's technique uses test functions w_i that are identical to the shape functions s_i. In the SUPG technique, the test function w_i is obtained from s_i by generalizing formula [8.26] to multiple dimensions as:

$$w_i = s_i + a_i \vec{\lambda}.\overrightarrow{\text{Grad}s_i} \qquad [8.48]$$

where the vector $\vec{\lambda}$ is formed by the components λ_x, λ_y and $\overrightarrow{\text{Grad}}$ denotes the gradient operator. Applying equation [8.48] to equation [8.46] yields:

$$w_i(x,y) = 1 + \lambda_x E_{i,j,k} + \lambda_y F_{i,j,k} + E_{i,j,k}(x - x_i) + F_{i,j,k}(y - y_i) \qquad [8.49]$$

As in the one-dimensional case, the test function w_i gives an increased weight to the nodes upstream of node i in the discretized equation, while the weight of the nodes downstream of i is reduced.

8.4. Discontinuous Galerkin techniques

8.4.1. *Principle of the method*

The classical finite element techniques presented in sections 8.1 to 8.3 solve the weak forms of conservation laws. However, due to the non-uniqueness of weak solutions, conservation is not guaranteed in the general case. This is illustrated by the application examples presented in section 8.5, where the solution of simple, non-linear scalar laws such as the inviscid Burgers equation is shown to give erroneous shock propagation speeds.

Discontinuous Galerkin (DG) techniques allow conservation to be guaranteed via a slight modification of the weak form of the governing equations. They can be seen as a combination of finite volume and finite element techniques that retains the advantages of both methods. They were applied to hyperbolic conservation laws in [COC 89b], to one-dimensional hyperbolic systems in [COC 89a], multidimensional conservation laws in [COC 90], multidimensional systems in [COC 98] and convection problems in [COC 01]. Applications to the shallow water equations can be found in [DAW 02] and [KES 09], with an extension to two-dimensional transport in [AIZ 02, AIZ 03]. An application to morphological modeling can be found in [KUB 06]. An improved slope limitation for one- and two-dimensional systems can be found in [GHO 09]. The method is presented for one-dimensional systems hereafter.

Assume that a hyperbolic system in conservation form [2.2], recalled here:

$$\frac{\partial \mathrm{U}}{\partial t} + \frac{\partial \mathrm{F}}{\partial x} = \mathrm{S}$$

is to be solved over the domain $\Omega = [0, L]$. As in the previous sections, equation [2.2] is multiplied by a test function w and integrated over the solution domain Ω:

$$\int_0^L \frac{\partial \mathrm{U}}{\partial t} w \,\mathrm{d}x + \int_0^L \frac{\partial \mathrm{F}}{\partial x} w \,\mathrm{d}x = \int_0^L \mathrm{S}w \,\mathrm{d}x \qquad [8.50]$$

Using integration by parts in the middle integral, equation [8.50] becomes:

$$\int_0^L \frac{\partial \mathrm{U}}{\partial t} w \,\mathrm{d}x + \left[w\mathrm{F}\right]_0^L - \int_0^L \frac{\partial w}{\partial x} \mathrm{F} \,\mathrm{d}x = \int_0^L \mathrm{S}w \,\mathrm{d}x \qquad [8.51]$$

As in finite volume techniques, the domain Ω is discretized into M computational cells. Writing equation [8.51] for the cell i leads to:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathrm{U}}{\partial t} w \,\mathrm{d}x + [w\mathrm{F}]_{x_{i-1/2}}^{x_{i+1/2}} - \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial w}{\partial x} \mathrm{F} \,\mathrm{d}x = \int_{x_{i-1/2}}^{x_{i-1/2}} \mathrm{S}w \,\mathrm{d}x \qquad [8.52]$$

where the square brackets denote the variation of the function between the brackets, $[f]_a^b = f(b) - f(a)$. The solution U in the cell i is sought in the form:

$$\tilde{\mathrm{U}}_i^n(x) = \sum_{p=0}^{P} (\mathrm{U}_p)_i^n \, s_p(x) \qquad [8.53]$$

where the $s_p(x)$ are shape functions and $(\mathrm{U}_p)_i^n$ is a constant vector. P is fixed arbitrarily. Classically, the shape functions are polynomials in x. In this case, the method is said to be of order $P + 1$ (that is, $P = 0$ yields a first-order method, $P = 1$ yields a second-order method, etc.). Substituting equation [8.53] into equation [8.52] and using a first-order approximation for the time derivative leads to:

$$\sum_{p=1}^{p} \int_{x_{i-1/2}}^{x_{i+1/2}} s_p w \, dx \frac{(\mathrm{U}_p)_i^{n+1} - (\mathrm{U}_p)_i^n}{\Delta t} = -[wF]_{x_{i-1/2}}^{x_{i+1/2}} + \int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{\partial w}{\partial x} F + \mathrm{S}w \right) dx \qquad [8.54]$$

The solution is known completely over the cell i if all the coefficients $(\mathrm{U}_p)_i^n$, $p = 1, \ldots, P$ can be computed. Therefore, it is necessary to define P different weighting functions w over the cell i in order to form a P×P system that can be solved uniquely for the coefficients $(\mathrm{U}_p)_i^n$. In the Galerkin approach, the test functions are identical to the shape functions. Consequently, equation [8.54] is rewritten successively for $w = s_0$, $w = s_1$, …, $w = s_P$. The following system is obtained:

$$\sum_{p=1}^{p} R_{p,j}(\mathrm{U}_p)_i^{n+1} = \sum_{p=1}^{p} R_{p,j}(\mathrm{U}_p)_i^{n} - \Delta t\,[s_j F]_{x_{i-1/2}}^{x_{i+1/2}} + \Delta t \int_{x_{i-1/2}}^{x_{i-1/2}} \left(\frac{\partial s_j}{\partial x} F + s_j \mathrm{S} \right) dx, \quad j = 1,\ldots,P \qquad [8.55]$$

with

$$R_{p,j} = R_{j,p} = \int_{x_{i-1/2}}^{x_{i+1/2}} s_p(x) s_j(x)\,\mathrm{d}x \qquad [8.56]$$

As in Godunov-type methods (see Chapter 7), the flux F at each interface $x_{i–1/2}$ is computed by solving a Riemann problem. The left and right states of the Riemann problem are defined from the reconstructions [8.53] over the cells $i – 1$ and i:

$$\left.\begin{aligned} \mathrm{U_L} &= \widetilde{\mathrm{U}}_{i-1}^{n}(x_{i-1/2}) \\ \mathrm{U_L} &= \widetilde{\mathrm{U}}_{i}^{n}(x_{i-1/2}) \end{aligned}\right\} \qquad [8.57]$$

The variation $[w_j\,\mathrm{F}]$ and the two integrals on the right-hand side of equation [8.55] must be estimated. Their estimate, as well as the computation of the coefficients $R_{p,j}$, is examined in the following section.

8.4.2. *Legendre polynomial-based reconstruction*

Computation of the various terms in equations [8.55–8.56] is simplified if reconstruction [8.53] uses Legendre polynomials:

$$s_p(x) = \frac{1}{2^p} \sum_{k=0}^{p} \binom{p}{k}^2 \left(2\frac{x - x_i}{\Delta x_i} - 1 \right)^{p-k} \left(2\frac{x - x_i}{\Delta x_i} + 1 \right)^{k} \qquad [8.58]$$

where x_i is the abscissa of the center of the cell i. For $p = 0, 1, 2$, we have:

$$s_0(x) = 1, \quad s_1(x) = 2\frac{x - x_i}{\Delta x_i}, \quad s_2(x) = 6\left(\frac{x - x_i}{\Delta x_i} \right)^2 - \frac{1}{2} \qquad [8.59]$$

Polynomials [8.58] have a number of interesting properties:

– Property (P8.1). The Legendre polynomials form a family of orthogonal functions over the interval $[x_{i-1/2}, x_{i+1/2}]$. Consequently, $R_{j,p} = R_{p,j}$ is non-zero only if $j = p$.

– Property (P8.2). The polynomial takes particular values $s_p(x_{i-1/2}) = (-1)^p$ and $s_p(x_{i-1/2}) = 1$.

– Property (P8.3). The integral of $s_p(x)$ over the cell i is zero for all p, except for $p = 0$. This property can be seen as a direct consequence of property (P8.1) by noting that the integral of $s_p(x)$ over the cell i is equal to $R_{0,p}$ by definition.

As a direct consequence of property (P8.3), the integral of the reconstructed profile over the cell i is given by:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{U}(x,t^n)\,\mathrm{d}x = \sum_{p=0}^{P} (\mathrm{U}_p)_i^n \int_{x_{i-1/2}}^{x_{i+1/2}} s_p(x)\,\mathrm{d}x \quad = \Delta x\,(\mathrm{U}_0)_i^n \qquad [8.60]$$

Using properties (P8.1) and (P8.2), equation [8.55] is rewritten as:

$$\begin{aligned}(\mathrm{U}_j)_i^{n+1} &= (\mathrm{U}_j)_i^n + \frac{\Delta t}{R_{j,j}}\left[(-1)^j \mathrm{F}_{i-1/2} - \mathrm{F}_{i+1/2}\right] \\ &+ \frac{\Delta t}{R_{j,j}} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{\partial s_j}{\partial x}\mathrm{F} + s_j \mathrm{S} \right) \mathrm{d}x, \qquad j = 1,\ldots,P\end{aligned} \qquad [8.61]$$

The integral in equation [8.61] may be estimated using for instance a second-order approximation (but other quadrature rules may be applied, see e.g. [KES 09]):

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(x)\,\mathrm{d}x \approx \frac{f(x_{i-1/2}) + 4f(x_i) + f(x_{i+1/2})}{6}\Delta x_i \qquad [8.62]$$

where f is defined as $f = \partial s_j / \partial x\,\mathrm{F} + s_j \mathrm{S}$. If an explicit formulation is chosen, f is easily determined at $x_{i-1/2}$, x_i and $x_{i+1/2}$ from the reconstructed profile of U at the known time level n. Indeed, from property (P8.2):

$$\left.\begin{aligned} \mathrm{U}_i(x_{i-1/2},t^n) &= \sum_{p=1}^{P}(-1)^p(\mathrm{U}_p)_i^n \\ \mathrm{U}_i(x_{i+1/2},t^n) &= \sum_{p=1}^{P}(\mathrm{U}_p)_i^n \end{aligned}\right\} \qquad [8.63]$$

The second-order DG method is obtained for $P = 1$. Writing equation [8.61] for $p = 0$ and $p = 1$ gives the following two formulae:

$$\left.\begin{aligned} (\mathrm{U}_0)_i^{n+1} &= (\mathrm{U}_0)_i^n + \frac{\Delta t}{\Delta x_i}\left(\mathrm{F}_{i-1/2} - \mathrm{F}_{i+1/2} + \int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{S}\,\mathrm{d}x\right) \\ (\mathrm{U}_1)_i^{n+1} &= (\mathrm{U}_1)_i^n + 3\frac{\Delta t}{\Delta x_i}\left[-\mathrm{F}_{i-1/2} - \mathrm{F}_{i+1/2} + \int_{x_{i-1/2}}^{x_{i+1/2}}\left(\frac{2\mathrm{F}}{\Delta x_i} + s_1\mathrm{S}\right)\mathrm{d}x\right] \end{aligned}\right\} \qquad [8.64]$$

where the integrals are approximated using e.g. equation [8.62]. Note that equations [8.64] are obtained using the following equalities:

$$\frac{\partial s_0}{\partial x} = 1,\ R_{0,0} = \Delta x_i,\ \frac{\partial s_1}{\partial x} = \frac{2}{\Delta x_i},\ R_{1,1} = \frac{\Delta x_i}{3} \qquad [8.65]$$

Remembering from equation [8.60] that $(\mathrm{U}_0)_i^n$ represents the average value of U over the cell i at time level n, the first equation [8.64] is formally equivalent to the finite volume equation [7.3]. In other words, the DG technique retains the conservation properties of finite volume methods.

8.4.3. *Limiting*

DG methods of arbitrary high order may produce oscillations in the computed variables near steep fronts. If the flux function F is nonlinear, the oscillatory character of the solution may lead to nonlinear instability. This can be avoided by limiting the variations in the reconstructed profiles so as to avoid under- or overshooting.

The coefficients $(\mathrm{U}_p)_i^n$ must be limited component by component. At the end of the limiting process, each component of the reconstructed edge value $\tilde{\mathrm{U}}_i^n(x_{i-1/2})$ should lie between the corresponding components of $(\mathrm{U}_0)_{i-1}^n$ and $(\mathrm{U}_0)_i^n$. Conversely, each component of the edge value $\tilde{\mathrm{U}}_i^n(x_{i+1/2})$ should lie between the corresponding components of $(\mathrm{U}_0)_i^n$ and $(\mathrm{U}_0)_{i+1}^n$.

If the cell i is a local extremum for a given component k, the profile for the kth component of the reconstructed profile is taken constant over the cell, that is, the kth component of $\tilde{\mathrm{U}}_i^n(x)$ is set to the value of the kth component of the vector $(\mathrm{U}_0)_i^n$. U_p must be limited as [KRI 07]:

$$\left[(\mathrm{U}_p)_i^n\right]_{\lim} = \operatorname{minmod}\left[(\mathrm{U}_p)_i^n, \frac{(\mathrm{U}_{p-1})_i^n - (\mathrm{U}_{p-1})_{i-1}^n}{2p-1}, \frac{(\mathrm{U}_{p-1})_{i+1}^n - (\mathrm{U}_{p-1})_i^n}{2p-1}\right] \qquad [8.66]$$

where the minmod function of two arguments a and b is zero if a and b have opposite signs, and retains the argument that has the smaller modulus if $ab > 0$:

$$\left.\begin{aligned} \operatorname{minmod}(a,b) &= \frac{\operatorname{sgn}(a)+\operatorname{sgn}(b)}{2}\min\left(|a|,|b|\right) \\ \operatorname{minmod}(a,b,c) &= \operatorname{minmod}\left[\operatorname{minmod}(a,b),c\right] \end{aligned}\right\} \qquad [8.67]$$

Equation [8.66] is to be applied component-wise by descending values of p, from $p = P$ to $p = 1$. The limiting is stopped when the first value of p such that:

$$\left[(\mathrm{U}_p)_i^n\right]_{\lim} = (\mathrm{U}_p)_i^n \qquad [8.68]$$

is reached. For a second-order DG technique, equation [8.66] simplifies to:

$$\left[(\mathrm{U}_1)_i^n\right]_{\lim} = \min\bmod\left[(\mathrm{U}_1)_i^n, (\mathrm{U}_0)_i^n - (\mathrm{U}_0)_{i-1}^n, (\mathrm{U}_0)_{i+1}^n - (\mathrm{U}_0)_i^n\right] \qquad [8.69]$$

8.4.4. *Runge-Kutta time stepping*

Runge-Kutta Discontinuous Galerkin (RKDG) techniques are obtained as generalizations of the explicit method presented in the previous sections. In fact, equations [8.55] and its particular expressions [8.61] and [8.64] are provided for a first-order approximation of the time derivative. They can be recast in the form [6.73], recalled here:

$$\mathrm{U}_i^{n+1} = \mathrm{U}_i^n + \Delta t\ \mathrm{MU}_i^n$$

where M is a matrix operator as defined in section 6.5.2. Runge-Kutta time stepping may be applied as in equation [6.76], recalled here:

$$\mathrm{U}_i^{n+1} = \mathrm{U}_i^n + \sum_{k=1}^{M} \frac{\Delta t^k}{k!} \mathrm{M}^k \mathrm{U}_i^n$$

In the particular case of a second-order Runge-Kutta (RK2) stepping, the formula simplifies into:

$$\mathrm{U}_i^{n+1} = \mathrm{U}_i^n + \Delta t\ \mathrm{MU}_i^n + \frac{\Delta t^2}{2} \mathrm{M}^2 \mathrm{U}_i^n = \mathrm{U}_i^n + \Delta t\ \mathrm{M}\left(\mathrm{U}_i^n + \frac{\Delta t}{2} \mathrm{MU}_i^n\right) \qquad [8.70]$$

Noting that the quantity $\mathrm{U}_i^{n+1} = \mathrm{U}_i^n + \Delta t / 2\ \mathrm{MU}_i^n$ is obtained by applying the numerical method [6.73] over half a time step, equation [8.70] can be translated into the algorithmic form as follows:

– step 1: use the DG technique over half a time step to compute an intermediate value $\mathrm{U}_i^{n+1/2}$ at the intermediate time level *n* + 1/2:

$$\mathrm{U}_i^{n+1/2} = \mathrm{U}_i^n + \frac{\Delta t}{2} \mathrm{MU}_i^n \qquad [8.71]$$

– step 2: use the intermediate value $\mathrm{U}_i^{n+1/2}$ to update the estimate of the operator MU and proceed to the next time level:

$$\mathrm{U}_i^{n+1} = \mathrm{U}_i^n + \Delta t\ \mathrm{MU}_i^{n+1/2} \qquad [8.72]$$

The second-order RKDG method is stable for a Courant number smaller than 1/3. The third-order RKDG method is stable for |Cr| smaller than 0.209, while the fourth-order RKDG method is stable for |Cr| smaller than 0.145 [COC 01].

8.5. Application examples

8.5.1. *The linear advection equation*

8.5.1.1. *Discretized equation*

In this section, the linear advection equation is solved using the Galerkin technique (equations [8.21–22]), the Petrov-Galerkin approach with piecewise constant test functions (equation [8.25]), the SUPG technique (equations [8.28–29]) and the DG technique (equations [8.64] with limiting [8.66–67] and RK2 time stepping [8.71–72]). The conservation and non-conservation forms of the linear advection equation are recalled:

$$\begin{aligned} &\frac{\partial U}{\partial t}+\frac{\partial}{\partial x}(\lambda U)=0 \qquad &&\text{(conservation form)} \\ &\frac{\partial U}{\partial t}+\lambda\frac{\partial U}{\partial x}=0 \qquad &&\text{(non - conservation form)} \end{aligned} \tag{8.73}$$

For the sake of simplicity, λ is assumed to be constant and positive over the solution domain. The element size Δx is taken as constant. Noticing that $C_{i,j}$ and $D_{i,j}$ are non-zero only for $j = i - 1$, $j = i$ or $j = i + 1$, equation [8.18] simplifies to:

$$\begin{aligned} &(C_{j-1,j}+\theta D_{j-i,j}\lambda\Delta t)U_{j-1}^{n+1}+(C_{j,j}+\theta D_{j,j}\lambda\Delta t)\ U_{j}^{n+1} \\ &+(C_{j+1,j}+\theta D_{j+i,j}\lambda\Delta t)\ U_{j+1}^{n+1}=\left[C_{j-1,j}-(1-\theta)D_{j-i,j}\lambda\Delta t\right]U_{j-1}^{n} \\ &+\left[C_{j,j}-(1-\theta)D_{j,j}\lambda\Delta t\right]U_{j}^{n}+\left[C_{j+1,j}-(1-\theta)D_{j+i,j}\lambda\Delta t\right]U_{j+1}^{n} \end{aligned} \tag{8.74}$$

Substituting equations [8.21–22] into equation [8.74] and dividing by Δx leads to a system in the form:

$$bU_{j-1}^{n+1}+cU_{j}^{n+1}+d\,U_{j-1}^{n+1}=e_{i}^{n} \tag{8.75}$$

where coefficients b, c, d and e are functions of the technique used.

– The SUPG approach with triangular shape functions leads to:

$$\left.\begin{aligned}
b &= \left[\frac{1}{6} + \frac{a\lambda}{2\Delta x} - \theta\left(\frac{1}{2} + \frac{a\lambda}{\Delta x}\right)\mathrm{Cr}\right] \\
c &= \left(\frac{2}{3} + \theta\,\mathrm{Cr}\right) \\
d &= \left[\frac{1}{6} - \frac{a\lambda}{2\Delta x} + \theta\left(\frac{1}{2} - \frac{a\lambda}{\Delta x}\right)\mathrm{Cr}\right] \\
e_i^n &= \left[\frac{1}{6} + \frac{a\lambda}{2\Delta x} + (1-\theta)\left(\frac{1}{2} + \frac{a\lambda}{\Delta x}\right)\mathrm{Cr}\right]U_{j-1}^n + \left(\frac{2}{3} - \theta\,\mathrm{Cr}\right)U_j^n \\
&\quad + \left[\frac{1}{6} - \frac{a\lambda}{2\Delta x} - \theta\left(\frac{1}{2} - \frac{a\lambda}{\Delta x}\right)\mathrm{Cr}\right]U_{j-1}^n
\end{aligned}\right\} \qquad [8.76]$$

where $\mathrm{Cr} = \lambda\,\Delta t/\Delta x$ is the Courant number and a is the stabilizing coefficient in equation [8.26]. Note that the Galerkin discretization is recovered with $a = 0$.

– The Petrov-Galerkin approach [8.25] leads to:

$$\left.\begin{aligned}
b &= \frac{1}{2} - \theta\,\mathrm{Cr} \\
c &= \frac{1}{2} + \theta\,\mathrm{Cr} \\
d &= 0 \\
e_i^n &= \left[\frac{1}{2} + (1-\theta)\,\mathrm{Cr}\right]U_{j-1}^n + \left[\frac{1}{2} - (1-\theta)\,\mathrm{Cr}\right]U_j^n
\end{aligned}\right\} \qquad [8.77]$$

– The second-order DG technique [8.64, 8.66–67, 8.71–72] is simplified by noting that the solution of the Riemann problem at the interface $i - 1/2$ for a positive λ is equal to the reconstructed value $\widetilde{U}_{i-1}^n(x_{i-1/2})$. Consequently:

$$\left.\begin{aligned}
F_{i-1/2} &= \lambda\widetilde{U}_{i-1}^n(x_{i-1/2}) = \left[(U_0)_{i-1}^n + (U_1)_{i-1}^n\right]\lambda \\
F_{i+1/2} &= \lambda\widetilde{U}_i^n(x_{i+1/2}) = \left[(U_0)_i^n + (U_1)_i^n\right]\lambda
\end{aligned}\right\} \qquad [8.78]$$

Moreover, we have:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{2F}{\Delta x_i}\,\mathrm{d}x = 2\lambda (U_0)_i^n \qquad [8.79]$$

Substituting equations [8.78] and [8.79] into equations [8.64] leads to:

$$\left.\begin{aligned} (U_0)_i^{n+1} &= (U_0)_i^n + \mathrm{Cr}\left[(U_0)_{i-1} + (U_1)_{i-1} - (U_0)_i - (U_1)_i\right] \\ (U_1)_i^{n+1} &= (U_1)_i^n + 3\mathrm{Cr}\left[-(U_0)_{i-1} + (U_0)_i - (U_1)_{i-1} - (U_1)_i\right] \end{aligned}\right\} \qquad [8.80]$$

NOTE.– Equation [8.75] may be written only at internal nodes ($i = 2, \ldots, M-1$). However, U_1^{n+1} is known from the upstream boundary condition, which adds an equation to the system. The missing equation for the downstream node ($i = M$) is supplied by applying the Petrov-Galerkin discretization [8.77].

8.5.1.2. *Test case description and results*

The following problem is solved: a constant value U_b is applied at the upstream end ($x = 0$) of the computational domain of length L. The initial value U_0 of U in the domain is uniform. The parameters of the test case are shown in Table 8.1, the computational results are illustrated by Figures 8.8 to 8.11. Note that the combination of λ, Δt and Δx adopted in this test yields a Courant number Cr = 2.

Recall that the Galerkin and Petrov-Galerkin techniques presented in this chapter are semi-implicit and that Courant numbers larger than unity can be handled without inducing stability problems.

Symbol	Meaning	Value
U_0	Initial value of U	0
U_b	Upstream boundary condition	1
L	Length of the domain	100 m
T	Simulated time	25 s
Δt	Computational time step	1 s
Δx	Cell size	1 m
λ	Wave speed	2.0 m/s
θ	Scheme implicitation parameter (Galerkin, Petrov-Galerkin and SUPG schemes)	0.50, 0.55, 0.65 and 1.00
ρ	SUPG's $a\lambda/\Delta x$ parameter	1

Table 8.1. *Linear advection. Parameters of the test case*

As shown in Figure 8.8, Galerkin's technique with triangular functions yields oscillations behind the computed front when θ is chosen close to 0.5. Increasing θ to 0.65 allows the oscillations to be eliminated almost completely.

For $\theta = 1.0$, the computed profile is monotonic, but the front is smeared over almost 15 cells. This was to be expected because increasing θ yields an increased numerical diffusion, the effect of which is to damp the shorter wavelengths that are responsible for the oscillations in the profile. The longer wavelengths are preserved, hence the smearing of the front.

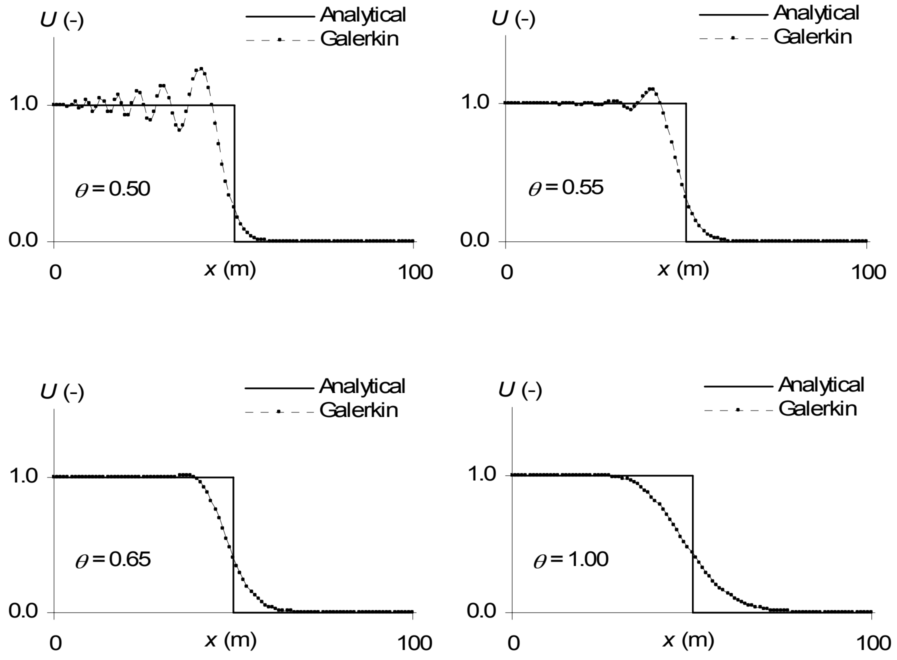

Figure 8.8. *Linear advection equation solved using Galerkin's technique. Numerical and analytical solutions at t = 25 s*

The Petrov-Galerkin technique [8.18, 8.25] gives similar results (Figure 8.9), except that the oscillations take place over a shorter distance than in the case of the Galerkin technique. They are almost damped for $\theta = 0.65$. The increased monotonic character of the Petrov-Galerkin technique is due to the stronger upwinding brought about by the piecewise constant test functions.

The SUPG technique (Figure 8.10) yields an increased damping of the oscillations compared to the Galerkin and Petrov-Galerkin discretizations. The oscillations behind the front are damped within a shorter distance than with the Petrov-Galerkin technique.

Owing to stability constraints, the Runge-Kutta discontinuous Galerkin technique is run with a much smaller time step than the first three methods. The time step chosen for the present application is $\Delta t = 0.125$ s, which corresponds to a Courant number Cr = 0.25 (remember that the stability limit is Cr = 1/3).

The profile computed at $T = 25$ seconds is compared with the analytical solution in Figure 8.11. Two numerical profiles are shown: the profile obtained without limiter and the profile obtained when limiting is applied. The profile obtained without limiting exhibits oscillations ahead of the front. In contrast, the limited profile is monotonic.

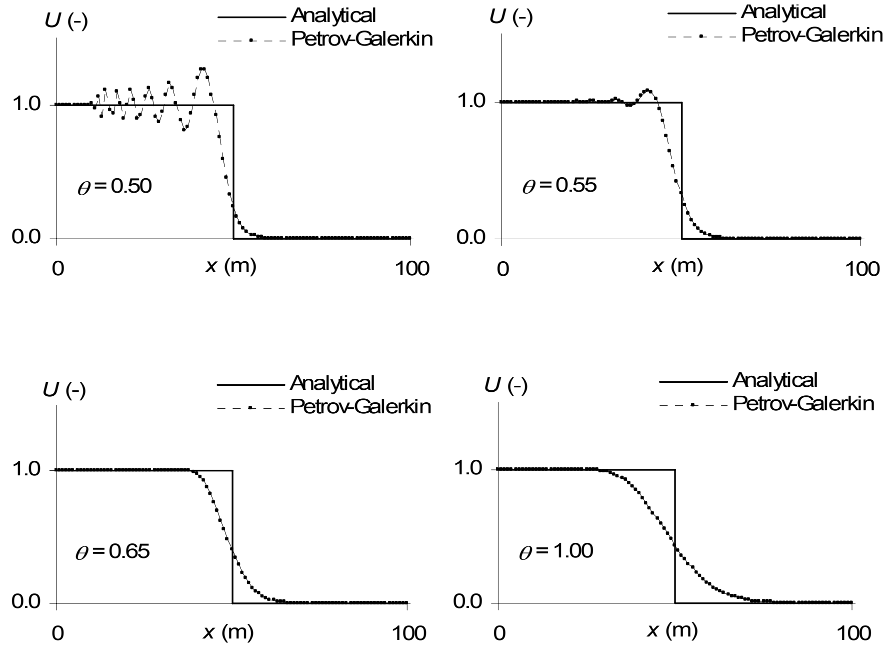

Figure 8.9. *Linear advection equation solved using the Petrov-Galerkin technique with piecewise constant weighting functions. Numerical and analytical solutions at t = 25 s*

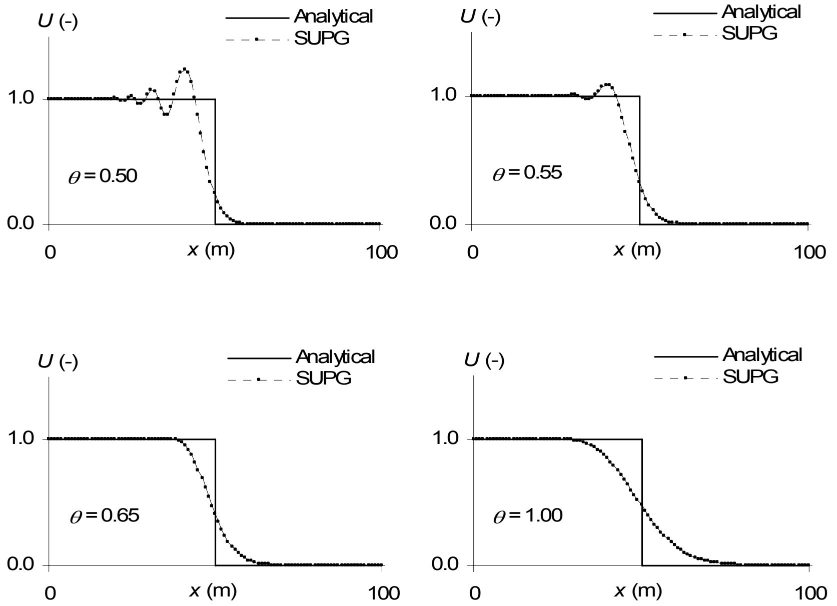

Figure 8.10. *Linear advection equation solved using the SUPG technique. Numerical and analytical solutions at t = 25 s*

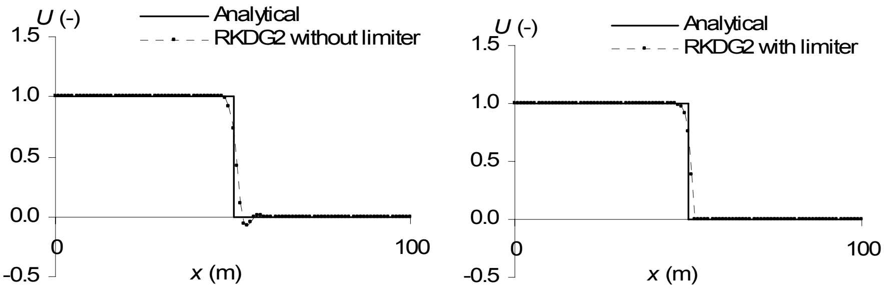

Figure 8.11. *Linear advection equation solved using the RKDG2 technique. Numerical and analytical solutions at t = 25 s*

8.5.2. *The inviscid Burgers equation*

8.5.2.1. *Solution by classical Galerkin techniques: explicit estimate of Cr*

The purpose is to solve the non-conservation form of the inviscid Burgers equation [1.66], recalled here:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

using the Galerkin, Petrov-Galerkin and SUPG techniques. The application example is limited to positive values of u.

Equations [8.76–77] may be used to compute the solution, provided that the wave propagation speed λ is replaced with u in the calculation of the coefficients b to e. The question then arises of how u, that is not uniform over the solution domain, should be estimated. In what follows, the following space weighting, inspired from Preissmann's scheme (see section 6.4.1), is used:

$$\mathrm{Cr} = \left[(1-\psi)u_{i-1}^{n} + \psi\, u_{i}^{n}\right]\frac{\Delta t}{\Delta x} \qquad [8.81]$$

Note that this expression is valid only for positive values of u and should be used in the computation of the coefficients for the node i.

The parameters used in the simulation are shown in Table 8.2.

Symbol	Meaning	Value
u_0	Initial condition	1 m/s
u_b	Prescribed velocity at the left-hand boundary	2 m/s
L	Domain length	100 m
T	Simulated time	50 s
Δt	Computational time step	1 s
Δx	Cell width	1 m
θ	Scheme implicitation parameter (Galerkin, Petrov-Galerkin and SUPG schemes)	0.50, 0.55, 0.65 and 1.00
ρ	Value of $a\lambda/\Delta x$ (SUPG scheme)	1.
ψ	Centering parameter for Cr in equation [8.81]	0.0, 0.5 and 1.0

Table 8.2. *The inviscid Burgers equation. Parameters of the test case*

The performance of the various schemes is illustrated in Figures 8.12 to 8.14. In these figures, the centering parameter in equation [8.81] is set to $\psi = 1/2$.

The solution computed by Galerkin's technique (Figure 8.12) exhibits strong oscillations for values of θ close to 0.5. We can check that for $\theta = 0.5$, the solution becomes unstable. Increasing θ allows the oscillations to be reduced and the shock speed to be better estimated. However, even the extreme value $\theta = 1.0$ does not allow the correct shock speed to be recovered in the numerical solution.

The Petrov-Galerkin method with a piecewise constant test function leads to a much smoother profile than the Galerkin technique (Figure 8.13). The shock speed is also better estimated. This, however, is achieved at the expense of a strongly smeared front.

The SUPG technique (Figure 8.14) combines the advantages of the previous two methods, with reduced oscillations compared to the Galerkin technique and a steeper front than in the Petrov-Galerkin technique. The drawback is a strongly underestimated shock speed in the numerical profile.

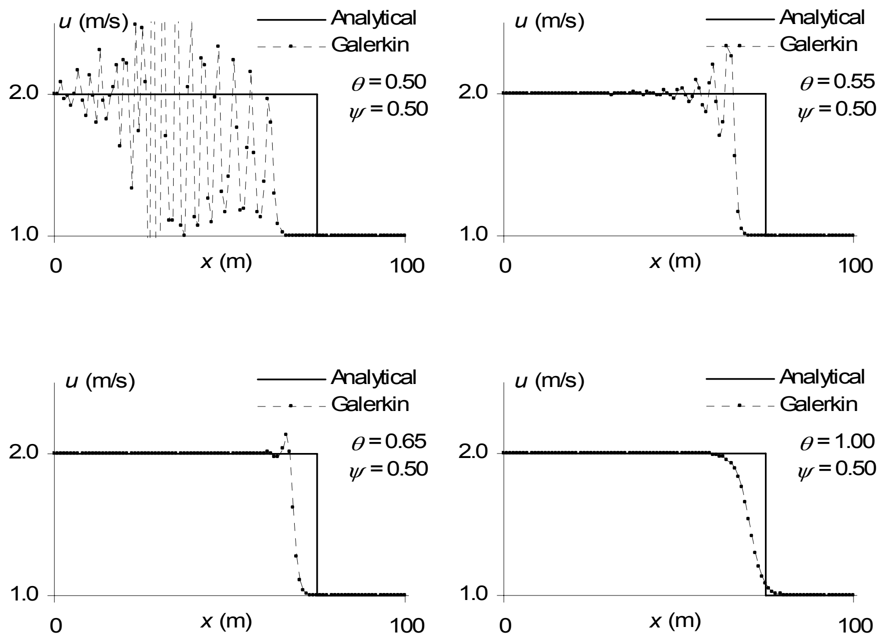

Figure 8.12. *The inviscid Burgers equation. Solution using Galerkin's technique at t = 50 s*

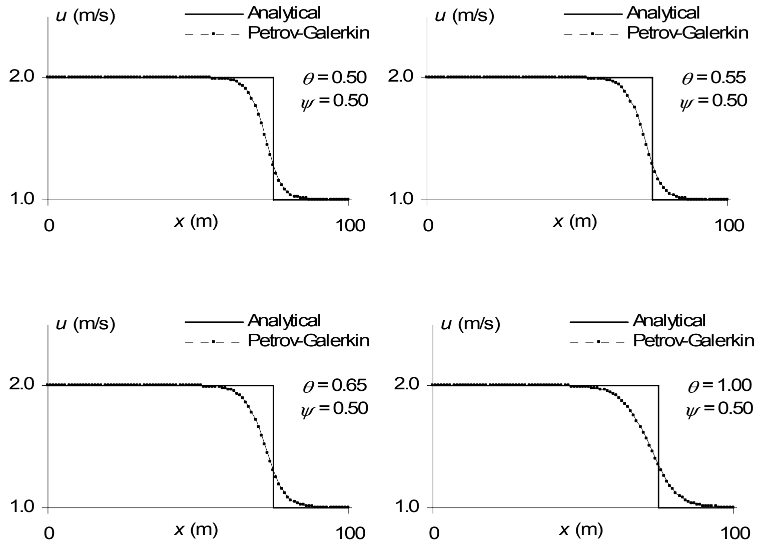

Figure 8.13. *The inviscid Burgers equation. Solution using the Petrov-Galerkin technique at t = 50 s*

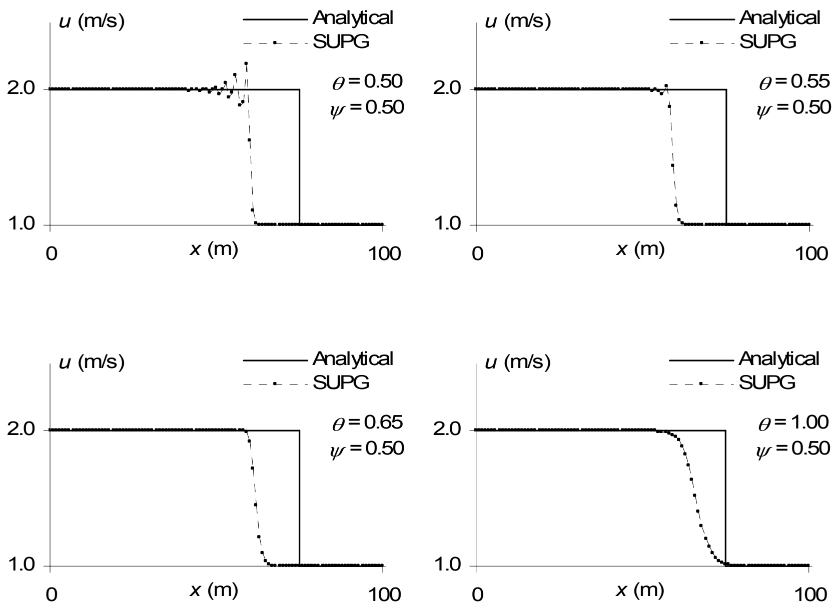

Figure 8.14. *The inviscid Burgers equation. Solution using the SUPG technique at t = 50 s*

The shock speed may be adjusted via the centering parameter ψ. In the present test case, u is larger on the left-hand side of the shock than on the right-hand side. Consequently, decreasing ψ in equation [8.81] should be expected to lead a larger interpolated value, and thus a larger shock speed. Conversely, increasing φ should be expected to yield a smaller shock speed.

This is confirmed by Figures 8.15 and 8.16, where the influence of ψ is studied for two different values of θ.

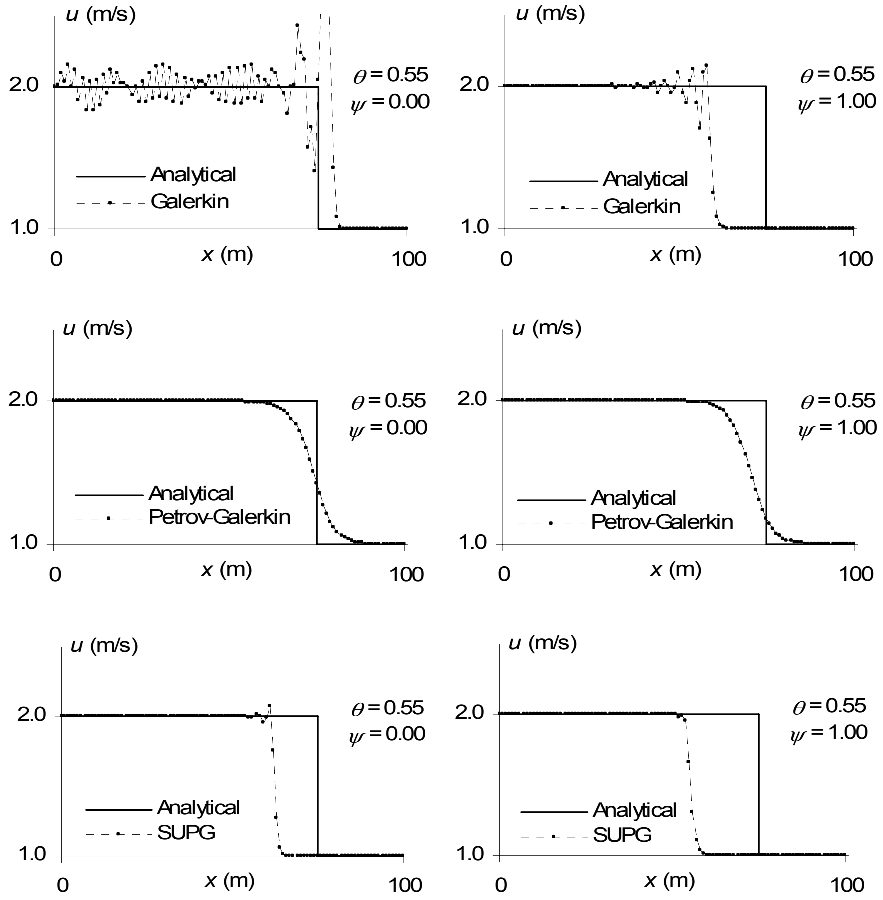

Figure 8.15b. *The inviscid Burgers equation. Influence of the centering parameter ψ for $\theta = 0.55$*

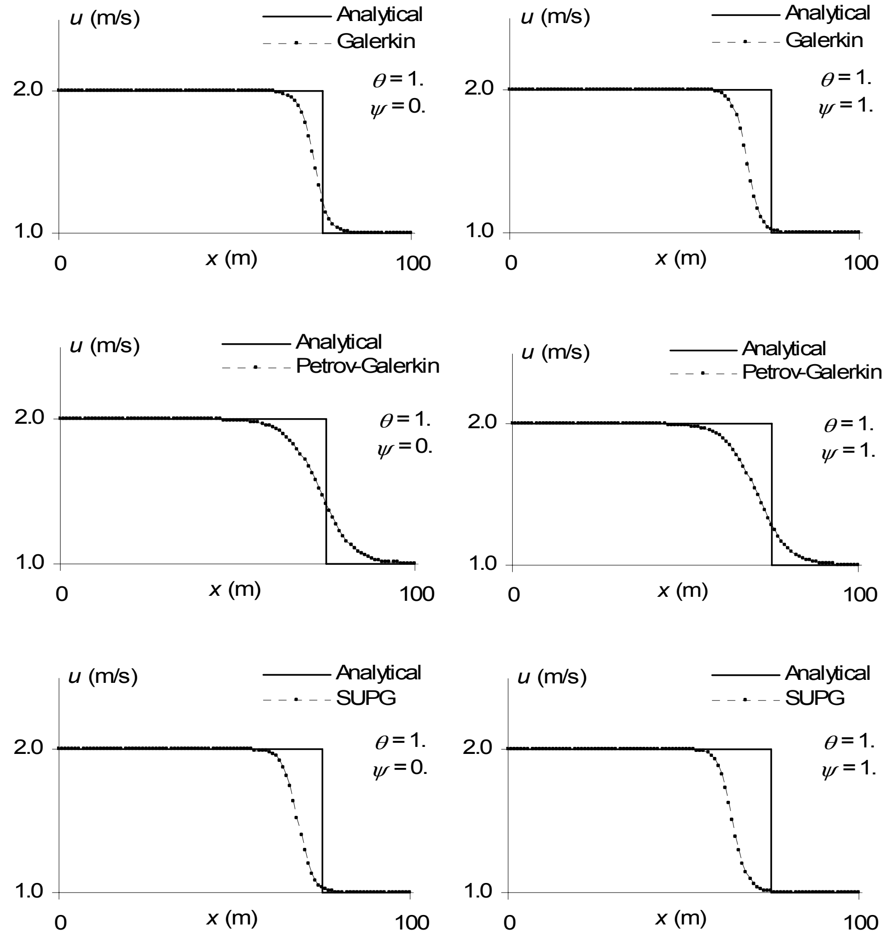

Figure 8.16. *The inviscid Burgers equation. Influence of the centering parameter ψ for θ = 1*

The Petrov-Galerkin technique allows the shock to be located more accurately than the other two methods, at the expense of an increased numerical diffusion. No combination of θ and ψ, however, allows any of the methods to locate the shock correctly. This militates in favor of a more accurate estimate of the Courant number. In the next section, a semi-implicit estimate is proposed.

8.5.2.2. *Solution by classical Galerkin techniques: semi-implicit estimate of Cr*

The following semi-implicit estimate is proposed for the Courant number:

$$\mathrm{Cr} = \Big\{(1-\theta_{\mathrm{Cr}})\Big[(1-\psi)u_{i-1}^{n} + \psi u_i^{n}\Big] + (\theta_{\mathrm{Cr}}\Big[(1-\psi)u_{i-1}^{n+1} + \psi u_i^{n+1}\Big]\Big]\Big\}\frac{\Delta t}{\Delta x} \qquad [8.82]$$

where θ_{Cr} is an implicitation parameter that is specific to the estimate of the Courant number. It does not necessarily take the same value as the parameter θ in the scheme in [8.76–77]. Equation [8.82] is a generalization of equation [8.81], that is obtained for $\theta_{\mathrm{Cr}} = 0$.

The semi-implicit estimate [8.82] is used within the following iterative procedure:

1) Assuming that the values at time level n are known, compute Cr for the first iteration using equation [8.81].

2) Use the thus estimated Cr to compute the coefficients in equations [8.76–77].

3) Solve the system of equations for the unknown values u_i^{n+1}. This provides a first estimate of the solution at the unknown time level $n + 1$.

4) Use the estimates u_i^{n+1} in equation [8.82] to update the estimate of the Courant number.

Steps 3) – 4) must be repeated until convergence is reached, that is, until two successive iterations yield the same (or almost the same) value for u_i^{n+1}. The appreciation of iteration convergence is left to the user of the method. In most practical cases, two or three iterations are seen to be sufficient.

The test case presented in section 8.4.2.1 is repeated. The parameters are given in Table 8.3.

As shown in Table 8.3, the iterative, semi-implicit estimate of the Courant number is a key factor in the accuracy of the method. For each of the techniques, it is possible to find a value for θ_{Cr} for which the shock is located correctly in the numerical solution. Indeed, in all three methods, $\theta_{\mathrm{Cr}} = 0$ (that is, the explicit estimate [8.79]) gives a shock that propagates too slowly, while $\theta_{\mathrm{Cr}} = 1.0$ yields a shock that propagates too fast.

The value $\theta_{\mathrm{Cr}} = 1/2$ gives good results for the Petrov-Galerkin technique. In the Galerkin and SUPG schemes, θ_{Cr} must be taken as slightly smaller than the value to recover a correct shock location, thereby guaranteeing mass conservation.

Symbol	Meaning	Value
u_0	Initial condition	1 m/s
u_b	Prescribed velocity at the left-hand boundary	2 m/s
L	Domain length	100 m
T	Simulated time	50 s
Δt	Computational time step	1 s
Δx	Cell width	1 m
θ	Scheme implicitation parameter	0. and 0.7
θ_{Cr}	Implicitation parameter for the estimate of Cr	0.5 and 1.0
ρ	Value if $a\lambda/\Delta x$ (SUPG scheme)	1.
ψ	Centering parameter for the estimate of Cr	0.5

Table 8.3. *Inviscid Burgers equation with semi-implicit estimate for Cr. Parameters of the test case*

These combinations of θ, θ_{Cr} and ψ must not be taken as general values that allow for conservation for all possible combinations of initial and boundary conditions. Consider the case of the Petrov-Galerkin technique with piecewise constant test functions [8.53].

As illustrated in Figure 8.17, the combination $\theta = 0.7$ and $\theta_{\mathrm{Cr}} = \psi_{\mathrm{Cr}} = 0.5$ guarantees conservation for $u_0 = 1$ m/s and $u_b = 2$ m/s. The test is repeated for the same combination of parameters, but the initial and boundary conditions are modified into $u_0 = 0$ m/s and $u_b = 5$ m/s. Figure 8.18 shows the computed profile at $T = 30$ s. It can be seen that the combination of numerical parameters is no longer optimal and does not guarantee conservation for this new set of initial and boundary conditions.

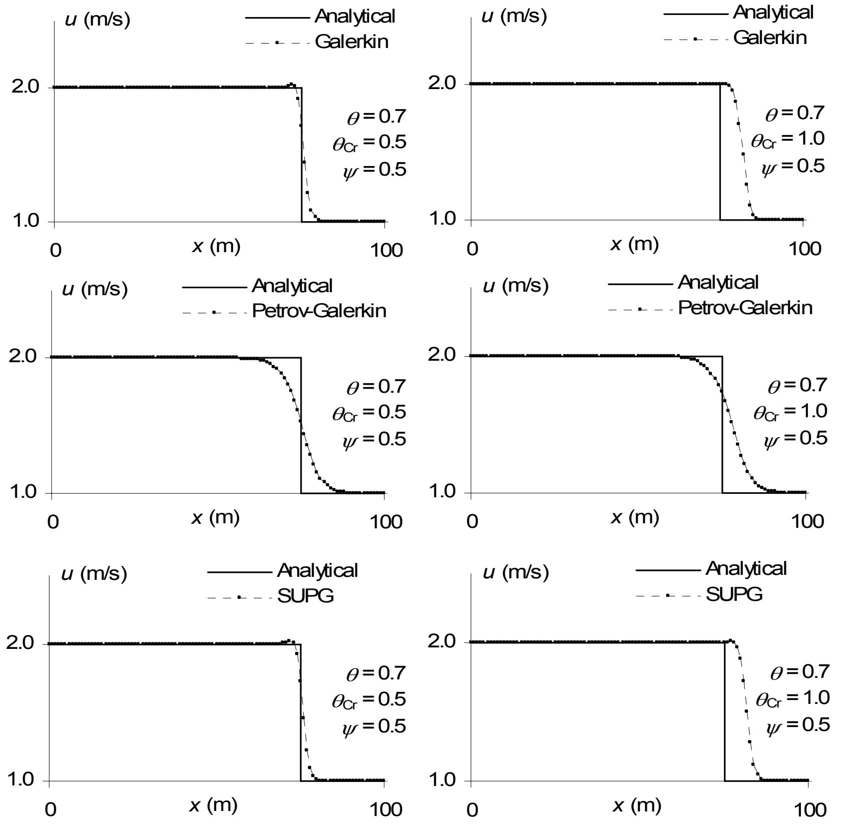

Figure 8.17. *The inviscid Burgers equation. Influence of the implicitation parameter θ_{Cr} for the Galerkin, Petrov-Galerkin with piecewise constant test function and SUPG techniques*

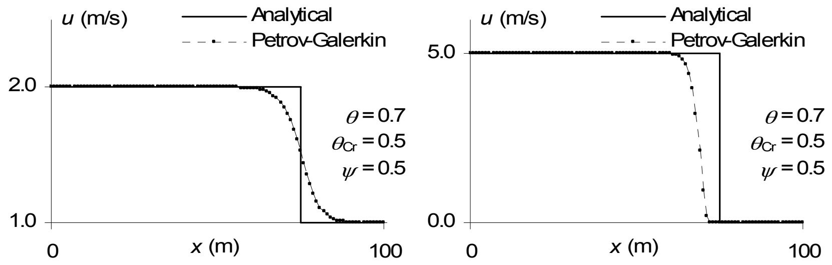

Figure 8.18. *The inviscid Burgers equation. Solution using the Petrov-Galerkin technique with piecewise constant weighting functions for two different combinations of initial and boundary conditions*

8.5.2.3. *Solution by the RKDG2 technique*

The RKDG2 technique is applied to the problem described in the previous sections. Figure 8.19 shows the numerical solution obtained at $T = 50$ seconds using the RKDG2 technique. The computational time step is $\Delta t = 0.1$ s. Obviously, conservation is ensured, at the expense however of computational rapidity, because the maximum permissible time step for solution stability is $\Delta t_{\max} = 0.15$ s.

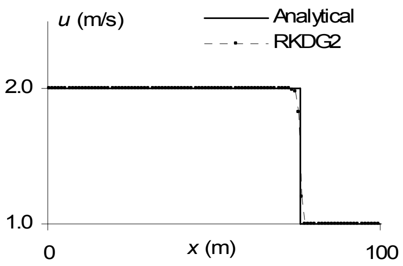

Figure 8.19. *The inviscid Burgers equation. Solution using the RKDG2 technique*

8.6. Summary

8.6.1. *What you should remember*

Finite element methods solve the weak form of conservation laws. The partial differential equation (PDE), or the system of PDEs, to be solved is multiplied by a pre-defined function, called a weighting or test function, and integrated over the solution domain. This means that the PDEs to be solved are solved in an average sense over the domain.

In classical Galerkin techniques (sections 8.1 to 8.3), space is discretized using computational points, also called nodes, that form elements. The solution is sought as the sum of pre-defined shape functions that take the value 1 at one node and 0 at all other nodes, multiplied by nodal values. The solution is known completely over the domain provided that each nodal value is known. In discontinuous Galerkin techniques (section 8.4), space is discretized using computational cells as in finite volume techniques. The solution is sought as a sum of elementary shape functions (for instance Legendre polynomials) defined over each cell, weighted by a cell coefficient. The solution is known over the computational domain provided that all the coefficients are known for all the elementary shape functions in each cell. In contrast with classical Galerkin techniques, discontinuous Galerkin techniques allow for discontinuous solutions at nodes.

Among the classical techniques, the Galerkin technique (see section 8.1.3.1) takes the shape and test functions from the same function space. In the Petrov-Galerkin technique (see section 8.1.3.2), the shape and test functions are taken from different function spaces. As a particular case, the SUPG approach (see section 8.1.3.3) uses test functions derived from the shape functions via the addition of a gradient-based term that allows for upwinding. This allows the oscillatory character of the numerical solution to be minimized to some extent.

When nonlinear PDEs (or systems) are to be solved in the framework of implicit or semi-implicit, classical Galerkin techniques, it is more convenient to solve the non-conservation form of the equations. However, this may lead to conservation problems, as shown by the computational examples of section 8.5.2. The example of the inviscid Burgers equation shows that a purely explicit estimate of the wave propagation speed gives incorrect shock speed estimates. The correct shock speed is recovered only if the computation of the wave propagation speed is made semi-implicit in the framework of an iterative procedure. In addition, a combination of implicitation and centering parameters that yield a correct shock speed for a given combination of initial and boundary conditions may give incorrect solutions for a different set of initial and boundary conditions.

Explicit, discontinuous Galerkin techniques introduced in section 8.4 are essentially conservative in that the computation of the average element value obeys a finite volume formalism. The higher-order components of the solution usually require limiting. Explicit discontinuous Galerkin techniques are stable when used in conjunction with Runge-Kutta algorithms, hence the initials RKDG for such methods. The stability constraints of RKDG techniques are more severe than those of classical finite volume schemes, thus constraining the permissible computational time steps to smaller values (see section 8.4.4).

8.6.2. *Application exercises*

Apply the Galerkin technique, the Petrov-Galerkin technique with piecewise constant test functions, the SUPG approach and the RKDG2 technique to the linear advection equation and the inviscid Burgers equation. Apply these discretizations to the test cases shown in section 8.5.

Indications and searching tips for the solution of these exercises can be found at the following URL: http://vincentguinot.free.fr/waves/exercises.htm.