2
Distributed Dynamic Models and Computational Fluid Dynamics

Young-il Lim and Sten Bay Jørgensen

2.1
Introduction

Chemical and biotechnical processes are often described by distributed dynamic models, that is, partial differential equations (PDEs) or partial differential algebraic equations (PDAEs) incorporating convection, diffusion, reaction and/or thermodynamic property terms. The PDAE models represent temporal as well as spatial variation of state variables. Since analytical solutions only exist in few cases, due to nonlinearity and complexity, computational methods (or numerical analyses) are generally required to solve such distributed dynamic models.

In this chapter numerical methods for solving PDEs are reviewed in the following three sections, first treating semidiscretized (method of lines) and fully discretized methods before discussing adaptive and moving mesh methods. Several applications of distributed models appearing in preparative chromatography, fixed-bed reactors, slurry bubble columns, crystallizers and microbial cultivation processes are treated in section 2.6 as a means to introduce various relevant aspects for the solution of PDE/PDAE models for chemical and biotechnical processes. Finally in section 2.7 an approach for combining computational fluid dynamic (CFD) technology with process simulation is illustrated and discussed.

2.2
Partial Differential Equations

Chemical and biotechnical processes often take place in spatially distributed systems and are therefore most appropriately described by distributed dynamic models, that is, partial differential equations (PDEs) incorporating convection, diffusion, reaction and thermodynamic property terms (Heydweiller et al. 1977; Kohler et al. 2001). For example, the material, energy, and momentum balances on moving fluid phases result in PDEs with respect to time and one or more space dimensions. Partial time derivatives occur as a direct consequence of the transient operation, while convective

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen
Copyright © 2006 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 3-527-30804-0

and diffusive (or dispersive) effects normally lead to first and second order partial space derivatives, respectively. Material and energy balances on stationary phases (e.g., the solid adsorbent in a packed-bed adsorber/reactor) may not involve any convective or diffusive terms and are therefore free of partial space derivatives. The properties of such a stationary phase at any single point obey ordinary differential equations (ODEs). Algebraic equations (AEs) are often used to define chemical equilibria, physical properties (e.g., enthalpy in terms of temperature, pressure and composition), or other intermediate quantities appearing in the differential equations. Therefore, physical models are generally expressed as PDEs coupled with AEs, i.e., so-called partial differential algebraic equations (PDAEs) submitted to initial conditions and boundary conditions.

For the purpose of this review, a PDAE system with one spatial coordinate can be expressed as follows:

$$\frac{\partial u}{t} = -\frac{\partial f(u)}{\partial x} + D\frac{\partial^2 u}{\partial x^2} + r(u, \theta) \tag{1a}$$

$$0 = g(u) \tag{1b}$$

where $u(t, x)$ is the state variable as a function of time ($t_0 \leq t \leq t_f$) and space ($x_0 \leq x \leq x_f$), $F(u)$ is the convection flux, D is the diffusion (or dispersion) coefficient, $r(u, \theta)$ is the reaction rate equation depending on state variables (u) and parameters (θ), and $g(u)$ is a nonlinear algebraic equation. On the right-hand side of Eq. (1a), the first, second and third terms take into account convection, diffusion and reaction, respectively. The partial differential equations govern a family of solutions. A particular member of the family of solutions is specified by the auxiliary conditions like initial and boundary conditions. For a PDE containing a first-order time derivative, one initial condition (IC) is required at an initial time level, $t = t_0$ along the space (x):

$$u(t_0, x) = u_0 \tag{2}$$

For a PDE containing a second-order spatial derivative like Eq. (1a), two boundary conditions are required at the physical boundaries of the solution domain. For example, the well-known Danckwert's boundary condition (BC) can be imposed for Eq. (1a):

$$f(u)\big|_{x=x_0} = f(u)\big|_{\text{in}} + D\frac{\partial u}{\partial x}\bigg|_{x=x_0} \quad \text{at } x = x_0 \text{ for all } t \tag{3a}$$

$$\frac{\partial u}{\partial x}\bigg|_{x=x_f} = 0 \quad \text{at } x = x_f \text{ for all } t \tag{3b}$$

where $f(u)||_{in}$ is the inlet flux predescribed by the operating condition. In the literature, Eq. (3b) is called the Neumann BC and Eq. (3a) is a mixture of the Dirichlet BC and the Neumann BC. Proper specification of auxiliary conditions is a necessary condition to obtain a well-posed problem (Hoffman 1993).

Physical mathematical models like the partial differential equations (PDEs) have a continuous form, while theose for solution purposes have to be discretized into a semidiscrete form (e.g., using only spatial discretization, Δx) or a fully discrete form (e.g., combining temporal and spatial discretization, Δt and Δx) in order to represent the models in the temporal and spatial (or computational) domain.

Among the large number of numerical methods developed for the solution of PDE or PDAE systems, the following is a well-established classification:

- Method of lines (MOL), including finite difference methods, finite element methods and finite volume methods (Finlayson 1980; Schiesser 1991; Leveque 1998; Lim et al. 2001a, Mantzaris et al. 2001a and 2001b).
- Fully discretized methods (Hoffmann 1993; Chang 1995 and 2002; Lim et al. 2004).
- Adaptive mesh refinement (AMR) or adaptive grid methods (Berger and Oliger 1984; Berger and LeVeque 1998; Vande Wouwer et al. 1998).
- Moving grid methods (Miller and Miller 1981; Dorfi and Drury 1987; Huang and Russell 1997; Li and Petzold 1997; Lim et al. 2001b).

The semidiscretized method is called MOL, where PDEs (or PDAEs) are converted into a system of ODEs (or DAEs) with respect to time by spatial discretization (see Section 2.3 for details). The main advantage is that well-established time integrators, e.g., Runge-Kutta or backward differentiation formula (BDF) methods, can be used for solving a large set of ODEs or DAEs. A main drawback is, however, that it is difficult to control and estimate the impact of the spatial discretization error (Oh 1995). For fully discretized methods (Section 2.4), a system of nonlinear algebraic equations is obtained after temporal and spatial discretization.

Adaptive and moving grid methods seem to be most promising since the idea is to use a numerical method in which nodes are automatically positioned in order to follow or anticipate steep moving fronts (Section 2.5). The node positioning may be achieved by using two basic strategies, namely AMR (i.e., local mesh refinement) and moving mesh methods (i.e., continuous spatial redistribution of a fixed number of mesh points). These two types of methods are appropriate for solving PDEs in the presence of steep moving fronts or discontinuities.

One of the key challenges facing process modeling today is the need to describe the interactions between fluid flow and phenomena models such as chemical reactions, mass transfer and phase equilibrium (Bezzo et al. 2003). Process simulations taking convection, diffusion and reaction into account and using computational fluid dynamics (CFD) for fluid hydrodynamics are important tools for the design and optimization of chemical and biochemical processes. The two technologies are largely complementary, each being able to capture and analyze some of the important process characteristics (Bezzo et al. 2000). Their combined application can therefore lead to significant modeling and simulation benefits, as will be discussed in Section 2.7. Before proceeding with this review, several preliminary concepts are summarized to facilitate presentation of the numerical methods.

2.2.1 ODE (or DAE) Integration

In the MOL framework, the numerical solution of PDEs (or PDAEs) is obtained by the time integration of the ODEs (or DAEs) resulting from spatial discretization. The general form of ODEs is expressed:

$$\mathrm{M}(t)\dot{u} = h(t, u) \tag{4}$$

where $\dot{\mathrm{u}}$ is the time derivative. When the matrix $\mathbf{M}(t)$ is singular, Eq. (4) represents a DAE system rather than an ODE system. Solving the DAE system is more complicated than solving the ODE system (see Section 2.2) because the DAE system only has a solution if the initial conditions u_0 are consistent in the sense that the equation $\mathbf{M}(t_0)\ \dot{\mathrm{u}}_0 = h(t_0, \mathrm{u}_0)$ has a solution, $\dot{\mathrm{u}}_0$, for the initial slope. Computations in a DAE integrator does not require $\mathbf{M}(t)$ to be nonsingular (Ascher and Petzold 1998).

If the time dependent ODE has a condition number that is large, then the problem is stiff. In other words, system (4) is stiff if the Jacobian $J = \frac{\partial}{h}\ \partial u$ (in the neighborhood of the solution) has eigenvalues λ_i, where $\frac{|\lambda_{\max}|}{|\lambda_{\min}|} >> 1$. For the stiff ODE/DAE systems, implicit BDF time integrators, such as DASSL (Petzold 1983), LSODI (Hindmarsch 1980), and DISCo (Sargousse et al. 1999) and *ode15s* in Matlab (The Mathworks Inc., MA, USA) are used for accurate evaluation of time derivatives.

2.2.2 Accuracy and Computational Performance

How close the numerical solution is to the true solution (or analytical solution, if that exists) at the finite temporal and spatial stepsizes (Δt and Δx) is assessed by evaluating the accuracy of the discretization. As Δt and Δx converge to zero and as the approximation order of derivatives increases, the approximation error generally diminishes. However, one must also account for computational efficiency in terms of the computational time that increases as the accuracy rises. In this context, there is a tradeoff between accuracy and computational efficiency. Thus, to simultaneously minimize the approximation error and the computational time, it can be considered as a multiobjective problem (Lim et al. 2001a).

The set of AEs or ODEs obtained after discretization of PDEs differs from the original PDE by the presence of the truncation error terms, which implicitly contribute to the numerical diffusion or dissipation. The truncation error related to accuracy is always present in the finite approximation of a PDE. An appropriate numerical method should be selected for a given PDE system in order to meet a tolerable numerical error within a reasonable computational time.

2.2.3 Automatic Differentiation

The numerical methods employed in the solution of many scientific computing problems require the computation of derivatives of some functions. Probably the best known are the gradient methods for optimization (e.g., successive quadratic programming, Powell 1971; see also Section 2.4), Newton's method for the solution of nonlinear algebraic equations (see Section 2.1), and the numerical solution of ODEs and PDEs. Both the accuracy and computational requirements of the derivative computation are usually of critical importance for the robustness and speed of the numerical solution (Bischof et al. 1992).

Taking a system of ODEs converted from a partial differential equation with MOL (see Section 2.1) as an example, the system is given as:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = h(t, \Delta x, u) \tag{5}$$

where the state variable $u = [u_1 \dots u_N$ and the nonlinear function $h = [h_1 \dots h_N$ are represented (or approximated) on N discrete spatial mesh points (or finite elements). ODE solution methods, such as implicit Runge-Kutta and BDF methods, require a $(N \times N)$ Jacobian $\left(\frac{\partial h}{\partial u}\right)$, which is either provided by the user or approximated by a difference quotient also called divided differences.

In fully discretized methods (e.g., the conservation element and solution element (CE/SE) method, see Section 2.2) for the numerical solution of a PDAE, a nonlinear system is obtained as a function of time, spatial stepsizes, and state variables.

$$0 = h(\Delta t, \Delta x, u) \tag{6}$$

For a fixed time and spatial stepsizes, Eq. (6) is solved by a Newton-type iteration requiring the Jacobian $\frac{\partial h}{\partial u}$. Therefore, the computation of derivatives (or Jacobian) is a crucial ingredient in the numerical solution of PDEs or PDAEs.

Hand-coding is increasingly difficult and error prone, especially as the problem complexity increases. Numerical approximation by divided differences has the advantage that the function is only needed as a black box. For example, a central divide difference is expressed as:

$$\left.\frac{\partial h_j}{\partial u_i}\right|_{u=u_0} \approx \frac{h_j(u_0 + \Delta u_i) - h_j(u_0 - \Delta u_i)}{2\Delta u_i} \tag{7}$$

The main drawback of divided differences is that their accuracy is difficult to assess. In addition, they are computationally expensive.

The basic idea of automatic differentiation (AD) is to avoid not only numerical approximations, which are expensive and contain rounding errors, but also hand-coded differentiation, which is error prone. Automatic differentiation techniques rely on the fact that every function, no matter how complicated, is evaluated on a computer as a sequence of elementary operations such as additions, multiplications and elementary functions (Bischof and Hovland 1991). By applying the chain rule

$$\frac{\partial}{\partial u_i} h_j\left(g(u)\right) = \frac{\partial h_j(g)}{\partial g} \cdot \frac{\partial g(u)}{\partial u_i} \tag{8}$$

over and over again to the composition of those elementary operations, one can compute derivative information of $h(u)$ exactly and in a completely mechanical fashion. Several AD packages such as Automatic Differentiation in FORTRAN (ADIFOR) (Bischof et al. 1998) are available from the AutoDiff organization Web site (http://www.autodiff.org/)).

2.2.4 Fixed, Adaptive and Moving Grids

The numerical study of evolutionary PDEs with steep moving fronts has demonstrated the need for numerical solution procedures with time and space adaptation. Over recent years, a great deal of interest has developed in adaptive mesh methods (Vande Wouwer et al. 1998). The objective of such approaches is to obtain solutions as accurately as could be obtained if a fine mesh was used over the entire physical domain, but at significantly lower computing cost. One would normally like to concentrate a large proportion of the nodes in regions where the solution exhibits rapid variation with respect to space. In the solution of many chemical engineering problems, steep moving profiles also appear. Common examples are (1) concentration breakthrough curves in fixed-bed absorbers (Kaczmarski et al. 1997), (2) particle (or crystal) size distribution governed by a population balance equation (Kumar and Ramkrishna 1997), and (3) heat conduction problems with a phase change (Mackenzie and Robertson 2000).

The fixed grid method uses the constant spatial mesh size (Δx) during time integration, whereas the moving mesh method continuously moves a fixed number of nodes to the regions of rapid solution variation over time. In the adaptive grid method (or AMR), meshes are locally added or removed at certain time levels according to solution steepness. In Section 2.3 adaptive and moving mesh methods are reviewed and compared.

2.3 Method of Lines

Time-dependent PDEs can be solved by means of the following two-stage procedures. First, the spatial variables are discretized on a selected spatial mesh so as to convert the original PDEs (or PDAEs) into a system of ODEs (or DAEs) with time as an independent variable. Secondly, the discretization in time of the ODE/DAE system then yields the required fully discretized scheme, normally using an ODE/DAE solver. This two-stage approach is often referred to as the method of lines (MOL) in the literature.

The spatial discretization means that the physical spatial domain ($x \in R^d$, in d dimensions) is discretized, replacing the analytical domain by its discrete equivalent domain (computational domain, $\xi \in R^d$) satisfying the original PDE in a finite number of discrete points distributed over the physical domain.

The discretization of PDEs on spatial domains normally leads to a Jacobian matrix whose elements lie within a narrow band (band Jacobian matrix). But, the bounded structure will be destroyed by the equations resulting from the boundary conditions, recycle streams or other nonlinear features. Hence, a sparse matrix would often be seen. In the ODE/DAE solver, the user will define the appropriate type of the Jacobian matrix to be evaluated by numerical difference, user-provided code or automatic differentiation, as discussed above.

The discretization techniques are important since not satisfying local conservation equations will give meaningless results. The numerical scheme has to closely mimic the behavior of the original PDEs and guarantee local conservation of flow properties. To achieve this, it is necessary to not only use conservative formulation of the governing equations but also a conservative numerical scheme.

The discretization of the spatial derivatives in Eq. (1) can be accomplished using three main categories: the finite difference method (FDM), the finite volume method (FVM) and the finite element method (FEM). The grid system may be a fixed grid, an adaptive grid or a moving grid.

2.3.1 Finite Difference Methods

The finite difference approximation of Eq. (1a) can be expressed in a simple way on N mesh points as follows:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{f(u_i) - f(u_{i-1})}{\Delta x} + D\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + r(u_i, \theta), \quad i = 2\ldots(N-1) \tag{9}$$

where the first-order spatial derivative is approximated by a first-order upwinding scheme under the condition $\frac{\partial f_i}{\partial u} \geq 0$ (or positive convective flow) and the second-order derivative by a central difference scheme. For $i = 1$ and N, boundary conditions such as Eq. (3) are applied.

The spatial discretization of the parabolic PDE (1) may cause stiffness due to second-order spatial derivatives, while that of the convection-dominated PDE (i.e., large convection velocity relative to diffusion coefficient) may cause instability, which is associated with oscillatory behavior of their solution due to first-order spatial derivatives (Finlayson 1980). The instability is encountered in using central schemes and higher-order upwinding schemes. To improve the accuracy of the FDMs for the PDEs, numerous attempts have been focused on approximating the first-order spatial derivatives, $\frac{\partial f}{\partial x}$ in Eq. (1a). Some guidance is provided in the selection of upwind methods in the FDM solution (Saucez et al. 2001).

In the traditional finite difference discretization (i.e., the fixed-stencil approach to be introduced later), the stencil (S_i) used to approximate spatial derivatives is fixed in both size (number of grid points) and position of the stencil points over the discretization procedure. Fixed-stencil (FS) approximations may not be adequate near discontinuities or steep fronts where they may give rise to oscillations. These problems have motivated the idea of an adaptive stencil (AS) (Shu and Osher 1989) and a weighted stencil (WS) (Jiang and Shu 1996). Namely, the left stencil shift (r) changes with the location x_i (see Fig. 2.1), but retaining the total number of points in the stencil.

We consider the cell-centered grid rather than vertex-centered grid, as shown in Fig. 2.1. Cells (C_i), cell centers (x_i), and stencils (S_i) in one spatial dimension are defined by

$$C_i = [x_{i-1/2}, x_{i+1/2}], \tag{10}$$

$$x_i = 0.5(x_{i-1/2} + x_{i+1/2}), \tag{11}$$

$$\begin{aligned} &S_i[C_{i-r}, C_{i-r+1}, \dots, C_i, C_{i+1}, \dots, C_{i+s}] = \\ &\quad S_i[x_{i-r-1/2}, x_{i-r+1/2}, \dots, x_{i-1/2}, x_{i+1/2}, \dots, x_{i+s-1/2}, x_{i+s+1/2}] \end{aligned} \tag{12}$$

where r and s denote the left and right stencil shifts, respectively. The approximation order (k) is defined as:

$$k - 1 = r + s \tag{13}$$

Consequently, FS approximations can be classified according to the left stencil shift (r) at the given k^{th}-order accuracy (see Table 2.1). For the AS methods (e.g., essentially nonoscillatory (ENO) schemes, Shu and Osher 1989), the left stencil shift (r) changes with locations (x_i) in order to avoid including a discontinuous (or steep front) cell (C_i) if possible. Just one stencil is selected out of some candidate stencils changed by r when doing the reconstruction, retaining the same order of accuracy.

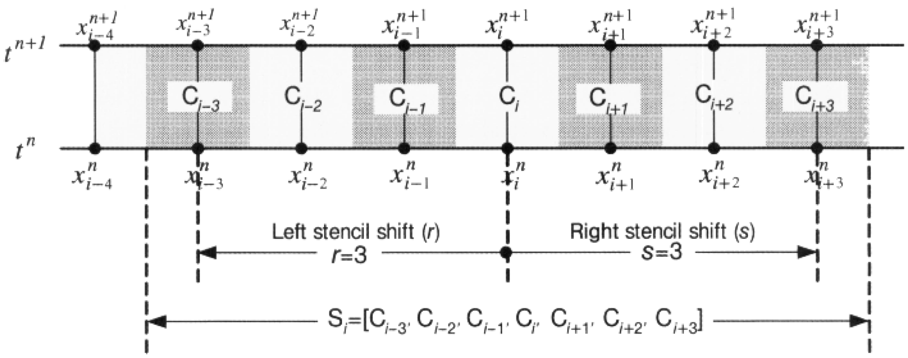


Figure 2.1 Stencil (S_i) and cell (C_i) structures in one dimensional problems

As a result, there are no oscillations and peaks are sharpened. The weighted stencil (Jiang and Shu 1996), however, uses all candidate stencils, each being assigned a nonlinear weight that depends on the local smoothness of the numerical solution.

For convective conservation laws, the one-dimensional hyperbolic PDE is expressed by,

$$u_t = -f_x \tag{14}$$

where the subscripts t and x indicate temporal and spatial partial derivatives ($\partial u/\partial t$ and $\partial f/\partial x$), respectively. If a function $h(x)$ satisfies at a discrete point x_i,

$$f_i(u) = \frac{1}{\Delta x_i}\int_{x-1/2}^{x+1/2} h(\xi)\,d\xi\,, \tag{15}$$

then, its derivative with respect to x (i.e., f_x) can be expressed as follows:

$$f_i(u)_x = \frac{1}{\Delta x_i}\left[h(x_{i+1/2}) - h(x_{i-1/2})\right] \tag{16}$$

Therefore, if a numerical flux $\hat{f}_{i+1/2}$ approximates $h(x_{i+1/2})$ to a k^{th}-order accuracy, the convection term can be discretized into k^{th}-order accurate conservative forms:

$$f_i(u)_x = \frac{\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right)}{\Delta x} \tag{17}$$

where $\hat{f}_{i+1/2}$ and $\hat{f}_{i-1/2}$ are numerical upflux and downflux, respectively, and the uniform mesh size (Δx) is used. Note that in the spatial direction, Eq. (17) can be considered as the finite volume discretization (see Section 2.3.3). The two numerical fluxes are exactly symmetrical with the one mesh distance. So, we only define the numerical upflux, $\hat{f}_{i+1/2}$ in this text.

2.3.1.1 Fixed-Stencil Approach

In the FS approach, the stencil (S_i) is fixed both in number and position. The numerical flux is approximated in a conservative manner from the flux point values with constants to meet k^{th}-order accuracy:

$$\hat{f}_{i+1/2} = \sum_{j=0}^{k-1} c_{rj} f_{i-r+j} \tag{18}$$

The constants c_{rj} are shown in Table 2.1 only for $r \geq 0$ (Shu 1997). Note that the constants c_{rj} are obtained from the derivative of the Newtonian interpolation polynomial (see Eq. (25) for details). For instance, FS-upwind-1 stands for a method of the fixed stencil with the first-order accuracy ($k = 1$) in the upwind direction. When the convection velocity is $\frac{\partial f}{\partial u} \geq 0$ in Eq. (14), its numerical upflux/downflux are given by Eq. (18) and Table 2.1:

Table 2.1 The constant c_{rj} up to fifth-order accuracy ($r \geq 0$)

Accuracy order (k)	Left stencil (r)	= 0	$j = 1$	$j = 2$	$j = 3$	$j = 4$	Reference name in this section
1	0	1					FS-upwind-1
2	0	1/2	1/2				FS-central-2
	1	−1/2	3/2				FS-back-2(TPB)
3	0	1/3	5/6	−1/6			FS-upwind-3
	1	−1/6	5/6	1/3			
	2	1/3	−7/6	11/6			
4	0	1/4	13/12	−5/12	1/12		FS-central-4
	1	−1/12	7/12	7/12	−1/12		
	2	1/12	−5/12	13/12	1/4		
	3	−1/4	13/12	−23/12	25/12		
5	0	1/5	77/60	−43/60	17/60	−1/12	FS-upwind-5
	1	−1/20	9/20	47/60	−13/60	1/30	
	2	1/30	−13/60	47/60	9/20	−1/20	
	3	−1/20	17/60	−43/60	77/60	1/5	
	4	1/5	−21/20	137/60	−163/60	137/60	

$$\hat{f}_{i+1/2} = c_{00} f_i = f_i, \quad \text{and} \quad \hat{f}_{i-1/2} = c_{00} f_{i-1} = f_{i-1} \tag{19}$$

so that,

$$f_x = \frac{(f_i - f_{i-1})}{\Delta x} \quad \text{for } \frac{\partial f}{\partial u} \geq 0 \tag{20a}$$

which was also introduced in Eq. (9). When the convection velocity is $\frac{\partial f}{\partial u}$ <0, Eq. (20a) is modified symmetrically in the opposite direction:

$$f_x = \frac{(f_{i+1} - f_i)}{\Delta x} \quad \text{for } \frac{\partial f}{\partial u} < 0 \tag{20b}$$

The first-order upwind scheme (FS-upwind-1) in Eq. (20) gives a very stable solution while poor accuracy because of low accuracy (k = 1). When k = 2 and r = 0, the FS-central-2 is obtained:

$$f_x = \frac{(f_{i+1} - f_{i-1})}{2\Delta x} \tag{21}$$

which is called the second-order central scheme. The FS-back-2 is equivalent to the three-point backward (TPB) method (Wu et al. 1990). In approximating the first-order spatial derivative, the central difference formulas (e.g., FS-central-2 and FS-central-4) tend to induce phase errors that appear in the form of numerical oscillations, as mentioned earlier. Higher-order upwinding schemes (e.g., FS-upwind-3 and FS-upwind-3 in Table 2.2) cannot remove the numerical oscillatory behavior in steep regions (Lim et al. 2001a).

2.3.1.2 Adaptive Stencil Approach

Finite difference ENO schemes were developed by Harten et al. (1987). They employed adaptive stencils in order to obtain information of solution gradients from smooth regions near discontinuities. This provides a sharp, ENO shock transition coupled with a formal uniformly high-order accuracy in smooth regions. Shu and Osher (1988, 1989) have proposed an efficient implementation of ENO schemes on the basis of fluxes rather than cell averages. The numerical fluxes $\hat{f}_{i+1/2}$ are evaluated using high-order interpolating polynomials constructed from adaptive stencils in the upwind direction.

The primitive of $h(x)$ from Eq. (15), $H(x)$, can be defined by:

$$H(x) = \int_{-x}^{x} h(\xi)\,\mathrm{d}\xi \tag{22}$$

$$H(x_{i+1/2}) = \sum_{j=-x}^{i} \int_{x_{j-1/2}}^{x_{j+1/2}} h(\xi)\,\mathrm{d}\xi = \sum_{j=-x}^{i} f_j \Delta x_j \tag{23}$$

Once $H(x)$ is approximated by a k^{th}-order Newtonian interpolation polynomial $P(x)$, using central divided differences (DD) at the $k+1$ points,

$$H(x) \approx P(x) = \sum_{j=0}^{k} DD\left[x_{i-r-1/2}, \ldots, x_{i-r+j-1/2}\right] \prod_{m=0}^{j-1} (x - x_{i-r+m-1/2}) \tag{24}$$

We can obtain the numerical flux $\hat{\mathrm{f}}_{i+1/2}$ through the derivative of the above equation.

$$\begin{aligned} \hat{f}_{i+1/2} &= \frac{\mathrm{d}P(x_{i+1/2})}{\mathrm{d}x} \\ &= \sum_{j=1}^{k} DD\left[x_{i-r-1/2}, \ldots, x_{i-r+j-1/2}\right] \sum_{m=0}^{j-1} \prod_{\substack{l=0 \\ l\neq m}}^{j-1} (x - x_{i-r+m-1/2}) \end{aligned} \tag{25}$$

The 0^{th} degree divided differences ($\mathrm{DDd}_i^{(0)}$) are defined by:

$$DD_i^{(0)}[x_{i-1/2}, x_{i+1/2}] = f_i \tag{26}$$

and in general the k^{th} degree divided differences, for $k \geq 1$, are defined by:

$$\begin{aligned} &DD^{(k)}[x_{i-r-1/2}, x_{i-r+1/2}, \ldots, x_{i-r+j-1/2}, x_{i-r+j+1/2}] = \\ &\frac{DD^{(k-1)}[x_{i-r+1/2} \ldots x_{i-r+j+1/2}] - DD^{(k-1)}[x_{i-r-1/2} \ldots x_{i-r+j-1/2}]}{x_{i-r+j+1/2} - x_{i-r-1/2}} \end{aligned} \tag{27}$$

For first-, second-, and third-degree divided differences, $DD^{1)}_{i+1/2}$, $DD_i^{(2)}$ and $DD^{3)}_{i+1/2}$ are defined, respectively, as flux point values (f_i):

$$DD^{(1)}_{i+1/2}[x_{i-1/2}, x_{i+1/2}, x_{i+3/2}] = \left(f_{i+1} - f_i\right)/2\Delta x \tag{28}$$

$$DD^{(2)}_{i}[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}] = \left(DD^{(1)}_{i+1/2} - DD^{(1)}_{i-1/2}\right)/3\Delta x \tag{29}$$

$$DD^{(3)}_{i+1/2}[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}, x_{i+5/2}] = \left(DD^{(2)}_{i+1} - DD^{(2)}_{i}\right)/4\Delta x \tag{30}$$

In the case of a fixed left stencil shift (r), Eq. (25) becomes equal to Eq. (18). Note that the constants c_{rj} (see Table 2.1) are obtained from Eq. (25), when the accuracy order (k) and the stencil shift (r) are given.

In the AS approach, the stencil shift (r) is adaptively chosen in Eq. (25). Since a smaller $|DD_i|$ implies that the function is smoother in that stencil, a smaller one (i.e., r) is chosen through the comparison of two relevant divided differences (e.g., $|DD_i|$ and $|DD_{i-1}|$). The ENO schemes are nonlinear even for linear problems and are especially suitable for problems containing both shocks and complicated smooth flow structures (Shu and Osher 1989).

The ENO schemes also have some drawbacks. One problem is the freely adaptive stencil, which could change by a rounding error perturbation near zeroes of the solution and its derivatives. Also, this free adaptation of stencils is not necessary in regions where the solution is smooth. Another problem is that the ENO schemes are not cost-effective because the ENO stencil selection procedure involves many logical statements (i.e., if/then statements).

2.3.1.3 Weighted Stencil Approach

The weighted stencil (WS) method (i.e., WENO scheme) is an approach used to overcome the aforementioned drawbacks while keeping the robustness and high-order accuracy of ENO schemes. The idea of WENO is to use a convex combination of all candidate stencils instead of approximating the numerical flux by using only one of the candidate stencils (Jiang and Shu 1996).

For the third order WENO scheme in the upwind sense, two candidate stencils are used to define the numerical upflux ($\hat{f}_{i+1/2}$). That is, based on the FS approximation at $k = 2$ (see Table 2.1), two numerical fluxes (i.e., q_0 and q_1) from the two stencils (S_i and S_{i+1} for $r = 0$) are incorporated with the weighting:

$$\begin{gathered}\hat{f}_{i+1/2}\left(f_{i-1}, f_i, f_{i+1}\right) = \sum_{r=0}^{1} w_r q_r \left(f_{i-r+1}, f_{i-r}\right) = w_0 q_0 + w_1 q_1 \\ \text{where, } w_0 = \frac{\alpha_0}{\alpha_0 + \alpha_1} \text{ and } w_1 = \frac{\alpha_1}{\alpha_0 + \alpha_1}.\end{gathered} \tag{31}$$

q_r is obtained as in the FS approach of Eq. (18) for $k = 2$ in Table 2.1:

$$q_0 = \sum_{j=0}^{1} c_{0j} f_{i+j} = c_{00} f_i + c_{01} f_{i+1} = (f_i + f_{i+1})/2 \tag{32a}$$

$$q_1 = \sum_{j=0}^{1} c_{1j} f_{i+j-1} = c_{10} f_{i-1} + c_{11} f_i = (-f_{i-1} + 3f_i)/2 \tag{32b}$$

The question now is how to define the weighting parameters (α_0 and α_1), such that the ENO property is achieved. The weighting parameters are calculated by divided differences ($DD^{1)}{}_{i+1/2}$ from Eq. (28):

$$\alpha_0 = \frac{2/3}{\left(\varepsilon + \left(2 \times DD^{(1)}_{i+1/2}\right)^2\right)^p} \tag{33a}$$

$$\alpha_1 = \frac{1/3}{\left(\varepsilon + \left(2 \times DD^{(1)}_{i-1/2}\right)^2\right)^p} \tag{33b}$$

where ε is a small positive real number that is introduced to avoid the denominator becoming zero (often, $\varepsilon = 10^{-6} \sim 10^{-10}$). It is suggested in Jiang and Shu (1996) that the power $p = 2$ is adequate to obtain ENO approximations. If a flow speed $\frac{\partial f_{i+1/2}}{\partial x}$ is negative, the numerical flux is defined in the reverse order as $\hat{f}_{i+1/2}$ (f_{i+1}, f_i, f_{i-1}). Thus, the WENO scheme is a type of upwinding scheme.

The WENO scheme has the following properties: (1) it involves no logical statements, which do appear, however, in the basic ENO schemes, (2) the WENO scheme based on the $(k - 1)^{th}$ order ENO scheme is a k^{th} order approximation in smooth regions, (3) it achieves the ENO property by emulating ENO schemes at discontinuities, (4) it is smooth in the sense that the numerical flux $\hat{f}_{i+1/2}$ is a smooth function and (5) the WS method combines the FS method with the AS method. Hence, the WENO scheme (Jiang and Shu 1996) improves on the ENO scheme in robustness, smoothness of fluxes, convergence properties and computational efficiency (Shi et al. 2000).

2.3.1.4 Comparison of FS, AS and WS Approaches

Table 2.2 displays the formulation of 12 spatial discretization methods. The notation X–Y–k is used, where X stands for stencil type (FS, AS or WS), Y indicates the highlighted characteristics (upwind, central or backward), and k is the approximation (or accuracy) order.

Table 2.3 shows how the stencil structure (i.e., mesh points used for the numerical upflux) changes the position of a shock. In the FS and WS approaches, the position and number of the mesh points do not vary with the shock position. However, a stencil $S_i[x_{i-1}, x_i, x_{i+1}]$ of $\hat{f}_{i+1/2}$ for the WS scheme is composed of two substencils $S_{i-1/2}[x_{i-1},$

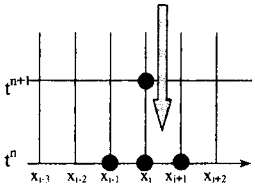
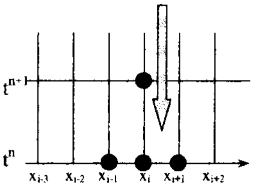
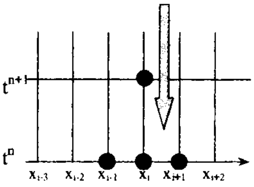
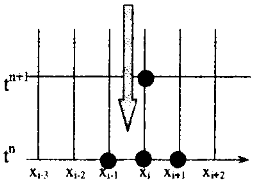
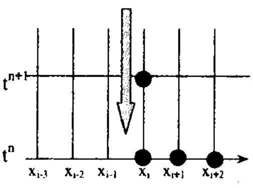
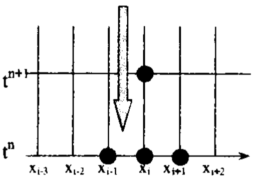
le 2.2 Classification of the flexible stencil methods (FS, AS and WS) in the upwinding sense, en the flow speed is positive

Fixed stencil (FS)		Adaptive stencil (AS)		Weighted stencil (WS)	
Name	**Spatial discretization (f_x)**	**Name**	**Numerical flux ($\hat{f}_{i+1/2}$)**	**Name**	**Numerical flux ($\hat{f}_{i+1/2}$)**
FS-upwind-1	$(f_i - f_{i-1})/h$*	–	–	–	–
FS-central-2 FS-back-2	$(f_{i+1} - f_{i-1})/2\,h$ $(3\,f_i - 4\,f_{i-1} + f_{i-2})/2\,h$	AS-upwind-2**	$\hat{f}^{(2)}_{i+1/2} = f_i + DD^{(1)}{}_{i\pm1/2}\,h$	–	–
FS-upwind-3	$(2\,f_{i+1} + 3\,f_i - 6\,f_{i-1} + f_{i-2})/6\,h$	AS-upwind–3	$\hat{f}^{(3)}_{i+1/2} = \hat{f}^{(2)}_{i+1/2}$ ..	WS-upwind-3	$[w_1(3\,f_i - f_{i-1}) + w_0(f_{i+1} + f_i)]/2$
FS-central-4	$(-f_{i+2} + 8\,f_{i+1} - 8\,f_{i-1} + f_{i-2})/12\,h$	AS-upwind-4	$\hat{f}^{(4)}_{i+1/2} = \hat{f}^{(3)}_{i+1/2}$ ..	WS-upwind-4	$[w_2(11\,f_i - 7\,f_{i-1}) + 2\,f_{i-2} + w_1(2\,f_{i+1} + 5$
FS-upwind-5	$(-3\,f_{i+2} + 30\,f_{i+1} + 20\,f_i - 60\,f_{i-1} + 15\,f_{i-2} - 2\,f_{i-3})/60\,h$	–	–	WS-upwind-5	$w_0(-f_{i+2} + 5\,f_{i+1} + 2\,f_i)]/6$

$h = \Delta x$

$DD^{(1)}_{i\pm1/2}$: first order divided differences, e.g., $DD^{(1)}_{i+1/2} = \frac{1}{2}\left(\frac{f_{i+1} - f_i}{x_{i+1} - x_i}\right)$, refer to Eq. (28).

Table 2.3 Stencil structures of third order FS/AS/WS methods for a numerical upflux ($\hat{f}_{i+1/2}$) in a positive flow velocity

	FS (FS-upwind-3)	AS (AS-upwind-3)	WS (WS-upwind-3)
Right shock	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}
Left shock	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}	 t^{n+1} t^n x_{i-3} x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2}

* The arrows indicate the position of shocks.

x] and $S_{i+1/2}[x, x_{i+1}]$ weighted with respect to the magnitude of neighboring divide differences (DD). In the third-order AS scheme in Table 2.3, the position of mesh points for the numerical upflux shifts to avoid a cell involving the shock. That is, the stencil adapts to solution variations.

To compare numerical performance for the three approaches, a linear equation of the conservation law is tested with an initial condition (u_0) of various wave forms:

$$u_t + u_x = 0, \quad -1.0 < x < 1.0, \tag{34}$$

$$u(x,0) = u_0(x) \tag{35}$$

where the initial condition

$$u_0 = \begin{cases} \left(e^{-\alpha(x+0.705)^2} + e^{-\alpha(x+0.695)^2} + 4e^{-\alpha(x+0.7)^2}\right)/6 & -0.8 \le x \le -0.6; \\ 1 & -0.4 \le x \le -0.2; \\ 1 - |10(x-0.1)| & 0.0 \le x \le 0.2; \\ \left(\sqrt{\max(1-100(x-0.495)^2, 0)} \right. & \\ \quad +\sqrt{\max(1-100(x-0.505)^2, 0)} & \\ \quad \left. +4\sqrt{\max(1-100(x-0.5)^2, 0)}\right)/6 & 0.4 \le x \le 0.6; \\ 0 & \text{otherwise} \end{cases}$$

contains a smooth but narrow combination of Gaussians $(e^{-\alpha(x-\beta)^2})$, a square wave, a sharp triangle wave and a half-ellipse (Jiang and Shu 1996). Since this PDE only has a linear convective term, its analytic solution shows the same shape as the initial condition, i.e., $u(x, t) = u_0(x - t)$.

Using a BDF ODE integrator (DISCo, Sargousse et al. 1999), the ODE system is solved on a PC. The band Jacobian matrix is numerically evaluated. To check accuracy at a given time level, the L_1-error is measured:

$$L_1\text{-error} = \int_{x=-1}^{x=1} \left| u(x)_{\text{analytical}} - u(x)_{\text{numerical}} \right| \, dx \tag{36}$$

Table 2.4 shows the benchmarking results achieved from the 12 discretization methods on uniform 200-mesh points (($\Delta x = 2/200$). The L_1-error is measured at $t = 0.4$ s and the computational time required is for time integration of 0.4 s. Instability is indicated by spurious oscillatory behavior in the numerical solutions.

Figure 2.2 depicts numerical performance within L_1-error vs. CPU time spaces from the data of Table 2.4. In general, as the approximation order increases, the error decreases. However, the fourth-order central discretization method (FS-central-4) produces much error over the FS-upwind-3 because of strong oscillations near the shock. Minimizing both the L_1-error and the computational time simultaneously, the six methods (FS-upwind-1, AS-upwind-3/4, and WS-upwind-3/4/5) are selected as the effective methods with consideration of the stability of the numerical solution (Lim et al. 2001a). It is found that the AS-upwind-2 takes an abnormally long computation time due to excessive iterations required for convergence.

In Fig. 2.3, the numerical solutions of FS-upwind-1 (shortest computational time) and WS-upwind-5 (smallest L_1-error) are compared to the FS-upwind-5. The numerical results of FS-upwind-1 are stable but not accurate due to the truncation error. The

Table 2.4 Accuracy, computational performance, and stability evaluation of the FDM in a linear convection equation ($N = 200$, L_1-error at $t = 0.4$ s and CPU time during 0.4 s integration time)

Stencil Type	Name	Accuracy (L_1 error)	Computational performance (CPU time, s)	Stability*
FS	FS-upwind-1	0.2696	0.8	O
	FS-central-2	0.1878	1.1	×
	FS-back-2	0.1345	1.1	×
	FS-upwind-3	0.0538	1.1	×
	FS-central-4	0.1219	1.3	×
	FS-upwind-5	0.0379	1.4	×
AS	AS-upwind-2	0.0961	9.1	O
	AS-upwind-3	0.0548	4.6	O
	AS-upwind-4	0.0440	4.8	O
WS	WS-upwind-3	0.0841	3.5	O
	WS-upwind-4	0.0452	4.9	O
	WS-upwind-5	0.0421	5.7	O

* Stability evaluation: O (stable), × (not stable)

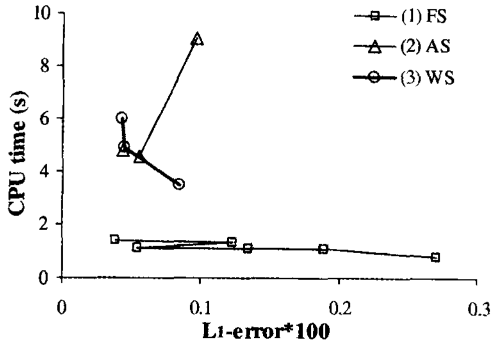


Figure 2.2 Error and computational time comparison of FS, AS and WS methods for the linear convection law (each curve follows the point from right to left). (1) FS: FS-upwind-1 → FS-central-2 → FS-back-2 → FS-upwind-3 → FS-central-4 → FS-upwind-5, (2) AS: AS-upwind-2 → AS-upwind-3 → AS-upwind-4, (3) WS: WS-upwind-3 → WS-upwind-4 → WS-upwind-5

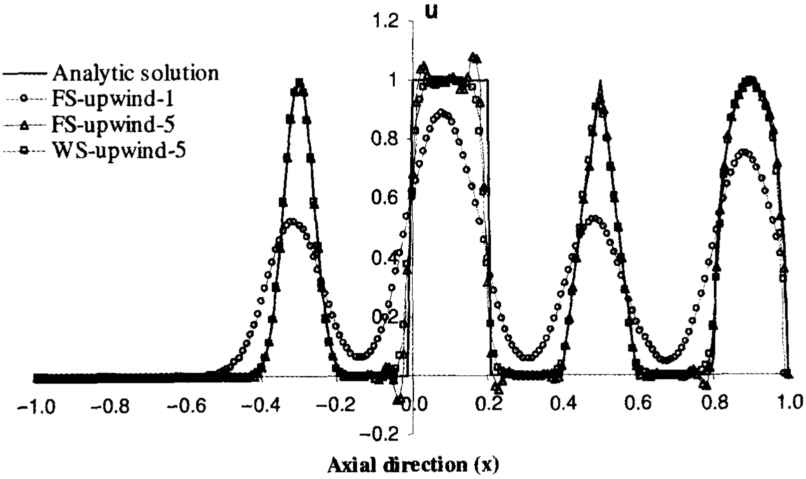


Figure 2.3 Numerical solutions of a linear convection equation according to discretization methods on 200 fixed-grid points at $t = 0.4$ s and $-1.0 \leq x \leq 1.0$

FS-upwind-5 method yields a stable solution in smooth regions, but produces some oscillations near discontinuities. One of the WS approaches, WS-upwind-5, is stable and accurate over all regions but computationally somewhat prohibitive (see Table 2.4).

2.3.2 Finite Element Methods

The finite element method (FEM) divides the physical domain into many smaller subdomains (elements) and applies weighted residual methods within each element. Each physical variable over the entire domain is expressed as a sum of finite elements. Additional restrictions are introduced to ensure various degrees of continuity of the solution at the element boundaries. In principle, any weighted residual methods can be combined with the finite element concept to yield a corresponding finite element method. In practice, the most commonly used method is the orthogonal collocation on finite elements (Finlayson 1980).

2.3.2.1 Orthogonal Collocation Method on Finite Elements

The method of orthogonal collocations on finite elements was presented in Villadsen and Michelsen (1978). An orthogonal collocation method approximates the solution by weighted combinations of orthogonal polynomials of degree M, and demands that the describing equations be satisfied exactly at a finite set of points called collocation points, which are the zeroes of an orthogonal polynomial. Table 2.5 lists the normalized collocation points for the orthogonal Legendre polynomials of degree of 2, 3 and 4 (Finlayson 1980).

Table 2.5 Normalized collocation points for orthogonal Legendre polynomials of degree 2, 3 and 4

Degree of polynomial	Collocation points
2	0, 0.5, 1
3	0, 0.21132, 0.78868, 1
4	0, 0.1127, 0.5, 0.8873, 1

In many areas, such as reaction engineering, the orthogonal collocation method has proved to be a powerful method leading to accurate results. However, when the solution has steep gradients, it is more beneficial to use it in conjunction with a finite element approach. As shown in Fig. 2.4, the physical domain is divided into a number of elements and an orthogonal collocation method is applied in each element. This gives rise to the orthogonal collocation method on finite elements.

The position of the j^{th} point in element i is denoted by x_{ij}. The approximated solution $\tilde{u}(x)$ in the element i can be given by:

$$\tilde{u}(x_i) \approx \sum_{j=0}^{M} \tilde{u}(x_{ij}) L_j^M(\hat{x}), \quad i = 1 \ldots N \tag{37}$$

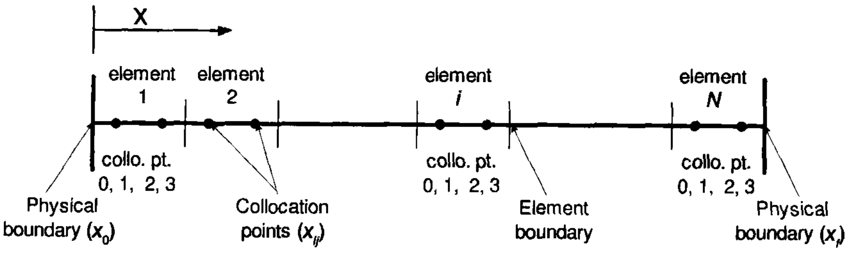


Figure 2.4 Grid system for orthogonal collocation on finite elements with polynomials of degree 3

where $L_j^M(\hat{x}) \equiv_{\substack{k=0\\k\neq j}}^{M} \dfrac{\hat{x}-\hat{x}_k}{\hat{x}_j-\hat{x}_k}$ is the Lagrange interpolation polynomial of degree M, N is the number of elements, and $\hat{x}$ is the normalized position within the element i:

$$0 \leq \hat{x} \equiv \frac{x_{ij}-x_{i0}}{\Delta x} \leq 1 \tag{38}$$

where Δx is the equidistant element length $\left(\Delta x = \dfrac{x_f - x_0}{N}\right)$.

From Eq. (37), the first-order derivative of the approximated solution $\tilde{u}(x)$ at position k in element i becomes:

$$\frac{\mathrm{d}\tilde{u}(x_{ik})}{\mathrm{d}x} \approx \frac{1}{\Delta x}\sum_{j=0}^{M}\tilde{u}(x_{ij})A_{jk}^{M}, \quad i = 1\ldots N,\ k = 0\ldots M \tag{39}$$

where A_{jk}^M is a constant $(M + 1) \times (M + 1)$ matrix defined by

$$A_{jk}^{M} \equiv \frac{\mathrm{d}L_j^M(\hat{x}_k)}{\mathrm{d}\hat{x}}, \quad j,k = 0\ldots M \tag{40}$$

From the definition of the Lagrange polynomial and the normalized collocation points in Table 2.5, the constants A^M *for* M *= 2, 3 and 4 are evaluated:*

$$A^2 = \begin{bmatrix} -3.0 & -1.0 & 1 \\ 4 & 0 & -4 \\ -1 & 1 & 3 \end{bmatrix} \tag{41a}$$

$$A^3 = \begin{bmatrix} -7 & -2.73205 & 0.73205 & -1 \\ 8.19615 & 1.73205 & -1.73205 & 2.19615 \\ -2.19615 & 1.73205 & -1.73205 & -8.19615 \\ 1 & -0.73205 & 2.73205 & 7 \end{bmatrix} \tag{41b}$$

$$A^4 = \begin{bmatrix} -13 & -5.32379 & 1.5 & -0.67621 & 1 \\ 14.78831 & 3.87298 & -3.22749 & 1.29099 & -1.87836 \\ -2.66667 & 2.06559 & 0 & -2.06559 & 2.66667 \\ 1.87836 & -1.29099 & 3.22749 & -3.87298 & -14.78831 \\ -1 & 0.67621 & -1.5 & 5.32379 & 13 \end{bmatrix} \tag{41c}$$

The second-order derivative of the approximated solution $\tilde{u}(x)$ at position k in element i can be obtained using a similar procedure:

$$\frac{\mathrm{d}^2\tilde{u}(x_{ik})}{\mathrm{d}x^2} \approx \frac{1}{\Delta x^2}\sum_{j=0}^{M}\tilde{u}(x_{ij})B_{jk}^{M}, \quad i = 1\ldots N,\ k = 0\ldots M \tag{42}$$

where B_{jk}^M is a constant $(M + 1) \times (M + 1)$ matrix defined by

$$B_{jk}^{M} \equiv \frac{\mathrm{d}^2L_j^M(\hat{x}_k)}{\mathrm{d}\hat{x}^2}, \quad j,k = 0\ldots M \tag{43}$$

Using Eq. (37), the formula for the integral of $\tilde{u}(x)$ over any one of the subintervals $|x_{ik}, x_{i,k+1}|$ can be derived as:

$$\int_{x_{ik}}^{x_{i,k+1}} \tilde{u}(x)\,\mathrm{d}x \approx \Delta x \sum_{j=0}^{M} \tilde{u}(x_{ij}) C_{jk}^{M}, \quad i = 1 \dots N,\ k = 0 \dots M-1 \tag{44}$$

where $C_{j\,k}^{M}$ is a constant $(M + 1) \times M$ matrix defined by:

$$C_{jk}^{M} \equiv \int_{\hat{x}_k}^{\hat{x}_{k+1}} L_j^M(\hat{x})\,\mathrm{d}\hat{x} \tag{45}$$

Using the orthogonal collocation method on finite elements, Eq. (1a) can consequently be approximated by:

$$\frac{\mathrm{d}\tilde{u}_{ik}}{\mathrm{d}t} = -\frac{1}{\Delta x} \sum_{j=0}^{M} f(\tilde{u}_{ij}) A_{jk}^{M} + \frac{D}{\Delta x^2} \sum_{j=0}^{M} \tilde{u}(x_{ij}) B_{jk}^{M} + r(\tilde{u}_{ik}), \quad i = 1 \dots N,\ k = 0 \dots M \tag{46}$$

It is interesting to note that the FDM and FEM presented here can be derived in a very similar fashion, namely by defining and manipulating interpolating polynomials over a finite set of points (see Eqs. (24) and (37)), despite their apparent differences. In both cases, spatial derivative approximations at a point x_i (or integral approximations over an interval $[x_{i-1/2}, x_{i+1/2}]$) involve the values of the function at a set of neighboring points. The main difference between the two methods is in the composition of the set. For the FDM, this normally involves a fixed number of points (i.e., stencil shift in Section 2.3.1) to the left and right of the current point x_i. For orthogonal collocation of the FEM, it involves all points within the element to which x_i belongs (Oh 1995).

2.3.2.2
Continuity at Element Boundaries

An important facet of all finite element methods is the treatment of the boundaries between elements. In general, the solution values are assumed to be continuous at the element boundaries, and this normally corresponds to physical reality. One could also make the first spatial derivative continuous across the interface, thus resulting in continuous solution approximations throughout the domain. However, in some cases (e.g., for inhomogeneous domains), it may be more appropriate to enforce continuity of some other quantity at the element boundaries, for instance, dispersive mass flux or conductive heat flux (Oh 1995).

The continuity of the first derivative at each boundary can be written as:

$$\left.\frac{\partial \tilde{u}(x_{iM})}{\partial x}\right|_{x=x_{iM}^-} = \left.\frac{\partial \tilde{u}(x_{iM})}{\partial x}\right|_{x=x_{iM}^+}, \quad i = 1 \dots N-1 \tag{47}$$

and can be described in the computational domain as follows:

$$\sum_{j=0}^{M} \tilde{u}(x_{ij}) A_{jM}^{M} = \sum_{j=0}^{M} \tilde{u}(x_{i+1,j}) A_{j0}^{M}, \quad i = 1 \dots N-1 \tag{48}$$

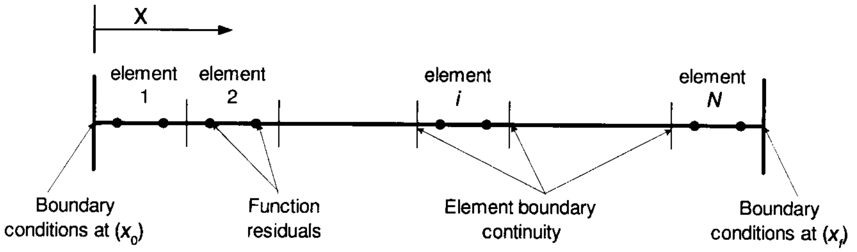

X
element 1
element 2
element i
element N
Boundary conditions at (x0)
Function residuals
Element boundary continuity
Boundary conditions at (xf)

Figure 2.5 Boundary conditions, residuals and element boundary continuity of orthogonal collocation on finite elements with polynomials of degree 3

In general, the discretization of a PDE and its associated boundary conditions using the orthogonal collocation method on finite elements results in three different classes of relation being applied at three different types of points (see Fig. 2.5):

- the appropriate boundary conditions are applied at the physical domain boundary such as Eq. (3) → 2 AEs (or 2 ODEs).
- the discretized PDE is enforced at the collocation points within each element → N(M – 1) ODEs
- element boundary continuity is enforced at the boundaries between elements → (N – 1) ODEs

Therefore, (NM + 1) DAEs (or ODEs) are obtained for one PDE.

2.3.3 Finite Volume Methods

For conservation laws, it is often preferable to use a finite volume method rather than a finite difference method in order to ensure that the numerical methods conserve the appropriate quantities of physical PDEs (Leveque 1998).

Consider the numerical solution of time dependent one-dimensional conservation laws,

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f(u)}{\partial x} = r(u) \tag{49}$$

for $(x, t) \in \Omega = (x_L, x_R) \times (0, T)$, where $r(u)$ accounts for all considered source and sink terms. We consider that the domain Ω is partitioned into strips such that

$$\in\Omega = \bigcup_{0\leq n\leq N_{\text{time}-1}} (x_L, x_R) \times (t^n, t^{n+1}) \tag{50}$$

where N_{time} is the number of time steps. Each strip is made up of two spatial grids in the case of nonuniform spatial grid, while in the fixed grid strips of control volume are rectangular as shown in Fig. 2.6.

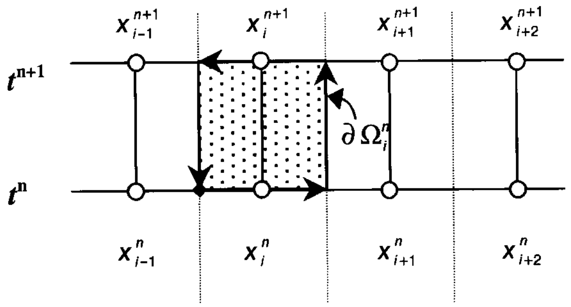


Figure 2.6 Time and space control volume filled with dots (Ω_i^n) and its path line ($\partial\Omega_i^n$) used on a uniform fixed grid

The midpoints of spatial grids ($x_{i+1/2}$) are defined at t^n simply as

$$x_{i+1/2}^n = \frac{1}{2}\left(x_{i+1}^n - x_i^n\right) \tag{51}$$

The finite volume approximation of Eq. (49) over the control volume Ω_i^n is derived from the original integral expression of the conservation law

$$\int_\Omega \left(\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x}\right) \mathrm{d}x\,\mathrm{d}t = \int_\Omega r(u)\,\mathrm{d}x\,\mathrm{d}t \tag{52}$$

Application of the divergence theorem based on Green's theorem[1] to Eq. (52) yields a line integral along the boundary, $\partial\Omega_i^n$. Performing the line integral on the left hand side of Eq. (52) yields:

$$\begin{aligned}\oint_{\partial\Omega_i^n} \left[f(u)\,Dt - u\,\mathrm{d}x\right] &= \int_{x_{i-1/2}^n}^{x_{i+1/2}^n} \left[f(u)\,\mathrm{d}t - u\,\mathrm{d}x\right] + \int_{t^n}^{t^{n+1}} \left[f(u)\,\mathrm{d}t = u\,\mathrm{d}x\right] \\ &+ \int_{x_{i+1/2}^{n+1}}^{x_{i-1/2}^{n+1}} \left[f(u)\,\mathrm{d}t - u\,\mathrm{d}x\right] + \int_{t^{n+1}}^{t^n} \left[f(u)\,\mathrm{d}t - u\,\mathrm{d}x\right]\end{aligned} \tag{53}$$

is obtained. For the right hand side, the source term is simply approximated:

$$\int_\Omega r(u)\,\mathrm{d}x\,\mathrm{d}t = \bar{r}(u)\Delta x_i \Delta t_n \tag{54}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$and $\Delta t_n = t^{n+1} - t^n$. An approximation of Eq. (53) is obtained using a numerical quadrature. A number of possibilities are available that give rise to either explicit or implicit methods. For example, using the midpoint formula to integrate along the bottom and top edges of Ω_i^n we get the approximations:

[1] 1 Green's theorem: Let R be a closed region bounded by C in the xy-plane. Let P(x,y) and Q(x,y) be functions defined and continuous first partial derivatives. Then

$$\iint_R \frac{\partial P}{\partial y}\,\mathrm{d}x\,\mathrm{d}y = -\oint_C P(x,y)\,\mathrm{d}x, \quad \iint_R \frac{\partial Q}{\partial x}\,\mathrm{d}x\,\mathrm{d}y = -\oint_C Q(x,y)\,\mathrm{d}y$$

$$\int_{x^n_{i-1/2}}^{x^n_{i+1/2}} f(u)\,\mathrm{d}t - u\,\mathrm{d}x \approx -\left(x^n_{i+1/2} - x^n_{i-1/2}\right) u^n_i = -\Delta x_i u^n_i\,, \tag{55}$$

$$\int_{x^{n+1}_{i+1/2}}^{x^{n+1}_{i-1/2}} f(u)\,\mathrm{d}t - u\,\mathrm{d}x \approx \left(x^{n+1}_{i+1/2} - x^{n+1}_{i-1/2}\right) u^{n+1}_i = \Delta x_i u^{n+1}_i \tag{56}$$

For the right and left hand side edges, we use the following family of approximation:

$$\int_{t^n}^{t^{n+1}} f(u)\,\mathrm{d}t - u\,\mathrm{d}x \approx \frac{1}{2}\left(f^n_{i+1/2} + f^{n+1}_{i+1/2}\right)\Delta t_n \tag{57}$$

$$\int_{t^{n+1}}^{t^n} f(u)\,\mathrm{d}t - u\,\mathrm{d}x \approx -\frac{1}{2}\left(f^n_{i-1/2} + f^{n+1}_{i-1/2}\right)\Delta t_n \tag{58}$$

Finally, the numerical results of the line integral yield:

$$\begin{aligned}\oint_{\partial\Omega^n_i} \left[f(u)\,\mathrm{d}t - u\,\mathrm{d}x\right] &= \Delta x_i\left(u^{n+1}_i - u^n_i\right)\\ &+ 0.5\left(f^n_{i+1/2} + f^{n+1}_{i+1/2} - f^n_{i-1/2} - f^{n+1}_{i-1/2}\right)\Delta t_n\end{aligned} \tag{59}$$

For the explicit form of f(u), approximating $f^{n+1}{}_{i+1/2}$ and $f^{n+1}{}_{i-1/2}$ to $f^n_{i+1/2}$ and $f^n_{i-1/2}$, respectively, we can obtain a simple numerical form of the conservation law as follows,

$$\frac{u^{n+1}_i - u^n_i}{\Delta t_n} = \frac{f^n_{i+1/2} - f^n_{i-1/2}}{\Delta x_i} + r\left(u^n_i\right) \tag{60}$$

which is a fully discrete formula for Eq. (49) where the convection term is discretized by the second-order central scheme (i.e., FS-central-2 in Section 2.3.1). Therefore, it seems that the explicit FVM for one-dimensional conservation law has almost the same formulation as the conservative FDM of Eq. (15). However, there are differences between the FVM and the FDM in accordance with the definition of the numerical fluxes $f^n_{i+1/2}$. In Eq. (60), $f^n_{i+1/2}$ is in fact an approximation to the average flux at $x = x_{i+1/2}$ along the finite volume Ω^n_i:

$$f^n_{i+1/2} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(u(x_{i+1/2}, t)\right)\,\mathrm{d}t \tag{61}$$

and for the conservative FDM at $t = t_n$, like Eq. (15):

$$f^n_{i+1/2} \approx \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} f\left(u(x, t_n)\right)\,\mathrm{d}x \tag{62}$$

Note that a complete FVM (namely the CE/SE method) in space and time domains is introduced in Section 2.4.3.

2.3.3.1 Spatial Finite Volume Method

Rather than attempting to discretize simultaneously in space and time, our attention is paid to discretization of spatial derivatives in the MOL using an adaptive time integrator, e.g., a BDF ODE solver. A naturally conservative spatial discretization procedure is provided by finite volume methods, where the discrete value f_i^n viewed as an approximation to the average value of $f(x_i, t^n)$ over a cell $C_i[x_{i-1/2}, x_{i+1/2}]$ in Fig. 2.1, rather than as an approximation to a point-wise value of $f(x_i, t^n)$ (Leveque 1998).

$$\frac{du}{dt} = -\frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x_i} + r(u_i) \tag{63}$$

where

$$\text{where } \hat{f}_{i+1/2} = \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} f(\xi)\, d\xi \tag{64}$$

The advantage of the semidiscretized approach is to achieve high accuracy in space. The cell average is simply the integration of f(x, t) over the cell divided by its area, so conservation can be maintained by updating this value based on fluxes through the cell edges. Although the derivation of such methods may be quite different from that of the conservative FDM, the resulting formulas are identical to Eq. (17).

The flux function $\hat{f}_{i+1/2}$ delivering high-order accuracy in space can be obtained by using higher-order interpolation polynomials like ENO schemes and WENO schemes (see Section 2.3.1).

2.4 Fully Discretized Method

The generic PDE with convection, diffusion and reaction terms can also be solved by temporal and spatial discretization of the original PDE. The time discretization procedure can be explicit or implicit. Several fully discrete schemes are introduced in finite difference and finite volume approximations.

2.4.1 Explicit Time Discretization

In this section we consider a PDE with the flow velocity (a) and diffusivity (D) like in Eq. (1a):

$$u_t = -au_x - Du_{xx} - r(u) \tag{65}$$

The above equation is discretized by the forward-time methods. For example, the Leapfrog scheme can be expressed on equidistant Δt and Δx as:

$$u_i^{n+1} = u_i^n - \frac{\nu}{2}\left(u_{i+1}^n - u_{i-1}^n\right) + \mu\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \Delta t r\left(u_i^n\right) \tag{66}$$

where $\nu = \dfrac{a\Delta \tau}{\Delta x}$ is called the Courant-Friedrichs-Lewy (CFL) number or convection number and $\mu = \dfrac{D\Delta t}{\Delta x^2}$ is the diffusion number (Hoffman 1993). The partial time derivative (u_t) is approximated by a first-order forward difference and the partial space derivatives (u_x and u_{xx}) are approximated by a second-order central difference. Since the central scheme of the first-order spatial derivative is unconditionally unstable as mentioned in Section 2.3.1, an upwind scheme is given by the equation below when $a \geq 0$:

$$u_i^{n+1} = u_i^n - \nu\left(u_i^n - u_{i-1}^n\right) + \mu\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \Delta t r\left(u_i^n\right) \tag{67}$$

The method is shown to be convergent but only conditionally stable. It introduces significant amounts of implicit numerical dissipation into the solution in the presence of steep fronts.

The Lax-Wendroff scheme (1960) is a very popular explicit finite difference method for hyperbolic PDEs (i.e., $D = 0$ in Eq. (65)). To suppress numerical instability caused by the central discretization, an artificial diffusion term is introduced by a second-order Taylor expansion in time:

$$u_i^{n+1} \approx u_i^n + (u_t)_i^n \Delta t + \frac{1}{2}(u_{tt})_i^n \Delta t^2 \simeq u_i^n - a(u_x)_i^n \Delta t + \frac{1}{2}a^2(u_{xx})_i^n \Delta t^2 \tag{68}$$

where the time derivatives u_t and u_{tt} are determined directly from $u_t = -au_x$. Applying central discretization to Eq. (68), Lax-Wendroff scheme is given for Eq. (65):

$$u_i^{n+1} = u_i^n - \frac{\nu}{2}\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{\nu^2}{2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \Delta t r\left(u_i^n\right) \tag{69}$$

From a stability analysis, the method is stable only if $|\nu| \leq 1$. However, it is not often used to solve convection-diffusion PDEs (Hoffman 1993). The Dufort-Frankel method (1953) proposed a modification of the Leapfrog scheme Eq. (66), which yields a conditionally stable explicit method. In this modification, u_i^n is replaced by the approximation $u_i^n = \dfrac{u_i^{n+1} - u_i^{n-1}}{2}$:

$$\begin{aligned}\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} &= -a\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right) + D\left(\frac{u_{i+1}^n - \left(u_i^{n+1} + u_i^{n-1}\right) + u_{i-1}}{\Delta x^2}\right) \\ &+ \frac{\Delta t}{2}\left(r\left(u_i^{n+1}\right) + r\left(u_i^{n-1}\right)\right)\end{aligned} \tag{70}$$

Solving Eq. (70) for u_i^{n+1} yields:

$$\begin{aligned}(1 + 2\mu)u_i^{n+1} &= -\nu\left(u_{i+1}^n - u_{i-1}^n\right) + (1 - 2\mu)u_i^{n-1} + 2\mu\left(u_{i+1}^n - u_{i-1}^n\right) \\ &+ \frac{\Delta t}{2}\left(r\left(u_i^{n+1}\right) + r\left(u_i^{n-1}\right)\right)\end{aligned} \tag{71}$$

The scheme cannot be used for the first time step because of the term u_i^{n-1}. The method is stable only if $|\nu| \leq 1$. However, large values of μ result in inaccurate solutions and a starting method is required to obtain the solution of the first time step.

The MacCormark method (1969) is based on the second-order forward time Taylor series as Eq. (68).

$$u_i^{n+1} \approx u_i^n + (u_t)_i^n \Delta t + \frac{1}{2}(u_{tt})_i^n \Delta t^2 \simeq u_i^n + \frac{1}{2}\left((u_t)_i^n + (u_t)_i^{n+1}\right)\Delta t \tag{72}$$

where $(u_t)_i^n$ is obtained from the approximation $(u)_i^n = (-au_i^n + D(u_x)^n{}_i)_x$. The method is composed of a predictor and a corrector step. In the predictor step, $\hat{u}_i^{n+1}$ is approximated by the first-order forward difference:

$$\hat{u}_i^{n+1} = u_i^n - \nu\left(u_{i+1}^n - u_i^n\right) + \mu\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \Delta tr(u_i^n) \tag{73}$$

For the corrector step, u_i^{n+1} is given:

$$u_i^{n+1} = \frac{1}{2}\left(u_i^n + \hat{u}_i^{n+1}\right) - \nu\left(\hat{u}_i^{n+1} - \hat{u}_{i-1}^{n+1}\right) + \frac{\mu}{2}\left(\hat{u}_{i+1}^{n+1} - 2\hat{u}_i^{n+1} + \hat{u}_{i-1}^{n+1}\right) + \Delta tr\left(u_i^n\right) \tag{74}$$

The two-sep method that shows second-order accuracy in both time and space is very popular for solving Eq. (65) and is conditionally stable.

A numerical solution can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as Δt and Δx go to zero. The necessary condition is called the CFL condition. All fully discrete explicit schemes require fulfillment of the CFL condition:

$$-1 \leq \nu \equiv \frac{a\Delta x}{\Delta t} \leq 1 \tag{75}$$

For stiff PDEs, implicit time discretization is usually preferred. The method of lines (see Section 2.3) using implicit time integrators is originally motivated to solve stiff PDEs, as mentioned above. In the next section, implicit methods fully discretized in both time and space are presented.

2.4.2 Implicit Time Discretization

The implicit Euler central difference method is:

$$u_i^{n+1} = u_i^n - \frac{\nu}{2}\left(u_{i+1}^{n+1} - u_{i-1}^{n+1}\right) + \mu\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) + \Delta tr\left(u_i^n\right) \tag{76a}$$

Rearranging Eq. (76a) yields:

$$-\left(\frac{1}{2}\nu + \mu\right)u_{i-1}^{n+1} + (1 + 2\mu)u_i^{n+1} + \left(\frac{1}{2}\nu - \mu\right)u_{i+1}^{n+1} = u_i^n + \Delta tr\left(u_i^n\right) \tag{76b}$$

Eq. (76) cannot be solved explicitly for $u^{n+1}{}_i$, because the two unknown neighboring values $u^{n+1}{}_{i+1}$ and $u^{n+1}{}_{i+1}$ also appear in the equation. Due to the implicit feature, this scheme, which has first-order accuracy in time and second-order accuracy in space, is unconditionally stable and convergent (Hoffman 1993). This implicit Euler method yields reasonable transient solutions for modest values of v and μ.

The Crank-Nicolson central difference scheme is constructed by a second-order approximation in both time and space:

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{v}{4}\left[\left(u_{i+1}^{n+1} - u_{i-1}^{n+1}\right) + \left(u_{i+1}^n - u_{i-1}^n\right)\right] \\ &+ \frac{\mu}{2}\left[\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) + \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)\right] \\ &+ \frac{\Delta t}{2}\left[r\left(u_i^{n+1}\right) + r\left(u_i^n\right)\right] \end{aligned} \tag{77}$$

The Crank-Nicolson method is also unconditionally stable and convergent.

The implicit nature of these methods yields a set of nonlinear algebraic equations, which must be solved simultaneously. Therefore, the iterative calculation requires substantial computational time, especially for multidimensional problems.

Recently, an explicit fully discrete method called the CE/SE method has been developed as a finite volume approach to solve fluid dynamics problems. The CE/SE method enforces flux conservation in space and time, both locally and globally. The method is explicit and, therefore, computationally efficient. Moreover, it is conceptually simple, easy to implement and readily extendable to higher dimensions. Despite its second-order accuracy in space, this method possesses low dispersion errors (Ayasoufi and Keith 2003).

2.4.3 Conservation Element/Solution Element Method

The CE/SE method has many nontraditional features, including a unified treatment of space and time, the introduction of conservation element (CE) and solution element (SE) and a novel shock capturing strategy without special techniques. Space-time CE/SE methods have been used to obtain highly accurate numerical solutions for 1D, 2D and 3D conservation laws involving shocks, boundary layers or contacting discontinuities (Chang 1995; Chang et al. 1999). The CFL number insensitive Scheme II (Chang 2002) has recently been proposed for the Euler equation (i.e., convection PDEs for mass, momentum and energy conservation). Stiff source term (e.g., a fast reaction) treatment for convection-reaction PDEs (Yu and Chang 1997) is also presented for the space-time CE/SE method. The extension to a PDAE system (Lim et al. 2004) derived from the original CE/SE method (Chang 1995) and Scheme II (Chang 2002) is presented in the following. Consider a PDE model like Eq. (65):

$$u_t = -f_x - p(u) \tag{78}$$

where the flux (f) implying convection and diffusion is defined as

$$f = au - Du_x \tag{79}$$

Thus, Eq. (78) is identical to Eq. (65). By the divergence theorem the equation is equal to flux conservation as follows:

$$\nabla \cdot h = p \tag{80}$$

where $\nabla = \left(\frac{\partial}{\partial x} \frac{\partial}{\partial t}\right)$ and $h = \left(\begin{matrix} f \\ u \end{matrix}\right)$. By using Gauss's divergence theorem (or Green's theorem) in a space-time E_2, it can be shown that Eq. (80) is the differential form of the integral conservation law:

$$\oint_{S(V)} h \cdot \mathrm{d}s = \int_V p \, \mathrm{d}V \tag{81}$$

where $S(V)$ is the boundary of an arbitrary space-time region V in E_2, and ds = d ($\sigma \cdot$ $\boldsymbol{n}$ with $d\sigma$ and $\boldsymbol{n}$, respectively, being the area and the outward normal vector of a surface element on $S(V)$. Note that, because $\boldsymbol{h} \cdot \boldsymbol{ds}$ is the space-time flux of $\boldsymbol{h}$ leaving the region V through the surface element $\boldsymbol{ds}$, Eq. (81) simply states that the total space-time flux of $\boldsymbol{h}$ leaving V through $S(V)$ is equal to the integral of p over V. Also, since in E_2, $d\sigma$ is the length of a differential line segment on the simple closed curve $S(V)$, the surface integral on the left-hand side of Eq. (81) can be converted into a line integral. In fact, Eq. (81) is equivalent to (Chang 1995):

$$\oint_{S(V)}^{c.c.} (-u \, \mathrm{d}z + f \, \mathrm{d}t) = \int_V p \, \mathrm{d}V, \tag{82}$$

where the notation c.c. indicates that the line integral should be carried out in the counterclockwise direction.

In Fig. 2.7, the mesh points (e.g., points A, C and E) are marked by circles. They are staggered in space-time. Any mesh point (j, n) is associated with a solution element $SE(j, n)$ and two conservation elements $CE_-(j, n)$ and $CE_+(j, n)$. By definition, $SE(j, n)$ is the interior of the shaded space-time region depicted in Fig. 2.7a. It includes a horizontal line segment, a vertical line segment, and their immediate neighborhood (Chang 1995). Also, by definition, (1) $CE_-(j, n)$ and $CE_+(j, n)$, respectively, are the rectangles ABCD and ADEF depicted in Fig. 2.7a and b; and (2) $CE(j, n)$ is the union of $CE_-(j, n)$ and $CE_+(j, n)$, i.e., the rectangle BCEF.

Let the coordinate of any mesh point (j, n) be (x_j, t^n) with $x_j = j\Delta x$ and $t^n = n\Delta t$. Then, for any $(x, t) \in SE(j, n)$, $u(x, t)$, $f(x, t)$ and $h(x, t)$, respectively, are approximated by a first-order Taylor expansion:

$$\tilde{u}(x_j, t^n) \equiv u_j^n + (u_x)_j^n (x - x_j) + (u_t)_j^n (t - t^n) \tag{83}$$

$$\tilde{f}(x_j, t^n) \equiv f_j^n + (f_x)_j^n (x - x_j) + (f_t)_j^n (t - t^n) \tag{84}$$

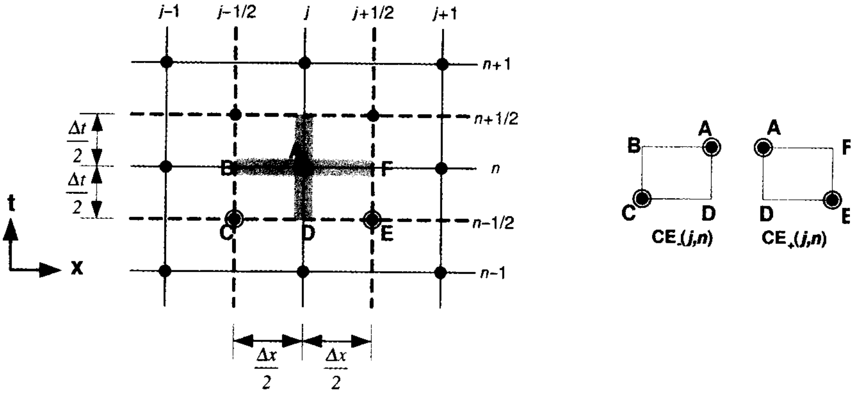


Figure 2.7 Solution element (SE) and conservation element (CE) at j^{th} position and n^{th} time level (Chang 1995). (a) Space-time staggered grid near SE(j, n). (b) CE$_-$(j, n) and CE$_+$(j, n)

so that,

$$\tilde{h}(x_j, t^n) \equiv \left(\tilde{f}(x_j, t^n), \tilde{u}(x_j, t^n)\right). \tag{85}$$

Here u_j^n, $(u_x)_j^n$, $(u_t)_j^n, f_j^n$, $(f^t)_j^n$ and $(f_t)_j^n$ are constants in $SE(j, n)$. In the CE/SE framework, $(u_t)_j^n, f_j^n$, $(f_x)_j^n$ and, $(f_t)^n{}_j$ are considered as functions of $(u)_j^n$ and $(u_x)_j^n$. These functions will be defined as follows. According to Eq. (79), one has:

$$f_j^n = au_j^n - D(u_x)_j^n \tag{86}$$

Also, by neglecting the contribution from the second-order derivative, $(f_x)_j^n$ may be obtained using the chain rule:

$$(f_x)_j^n = \left(\frac{\partial f}{\partial u}\right)_j^n (u_x)_j^n \approx a(u_x)_j^n \tag{87}$$

In order that $(u_t)_j^n$ can be determined in terms of $(u_x)_j^n$, it is assumed that for any $(x, t) \in SE(j, n)$,

$$\nabla \cdot \tilde{h}(x_j, t^n) = 0 \tag{88}$$

Thus, within $SE(j, n)$, the contribution of the source term (p) that appears in Eq. (80) is not modeled in Eq. (88). Note that (1) because it is the interior of a region that covers a horizontal line segment, a vertical segment and their immediate neighborhood, as shown in Fig. 2.7, $SE(j, n)$ is a space-time region with an infinitesimally small volume; and (2) as will be shown, the contribution of source terms will be modeled in a numerical analogue of Eq. (82). As a result, Eq. (88) implies:

$$(u_t)_j^n \approx -\left(f_x\right)_j^n \approx -a(u_x)_j^n, \tag{89}$$

$$(f_t)_j^n = \left(\frac{\partial f}{\partial u}\right)_j^n \left(\frac{\partial u}{\partial t}\right)_j^n \approx -\left(f_u\right)_j^n \cdot \left(f_u\right)_j^n \left(u_x\right)_j^n \approx -a^2(u_x)_j^n \tag{90}$$

Note that, by using Eqs. (86), (87), (89) and (90), f_j^n, $(f_x)_j^n$, $(u_t)_j^n$ and $(f_t)_j^n$ can be determined explicitly in terms of u_j^n and $(u_x)_j^n$.

2.4.3.1
Iterative CE/SE Method

The approximated conservation flux, $\tilde{F}_j^n$ in Eq. (82), is defined within $CE(j, n)$:

$$\tilde{F}_j^n \equiv \oint_{S(CE(j,n))}^{c.c.} \left(-\tilde{u}\,\mathrm{d}x + \tilde{f}\,\mathrm{d}t\right) \tag{91}$$

With the aid of Eq. (83) and (84), the line integral in Eq. (91) results in:

$$\tilde{F}_j^n = \frac{\Delta x}{2}\left[2u_j^n - u_{j+1/2}^{n-1/2} - u_{j-1/2}^{n-1/2} + s_{j+1/2}^{n-1/2} - s_{j-1/2}^{n-1/2}\right] \tag{92}$$

where

$$s_{j\pm1/2}^{n-1/2} \equiv \frac{\Delta x}{4}(u_x)_{j\pm1/2}^{n-1/2} + \frac{\Delta t}{\Delta x} f_{j\pm1/2}^{n-1/2} + \frac{\Delta t^2}{4\Delta x}\left(f_t\right)_{j\pm1/2}^{n-1/2} \tag{93}$$

The approximated source term flux ($\tilde{P}_j^n$) is obtained within $V(CE(j, n))$ as:

$$\tilde{P}_j^n \equiv \int_V (p)_j^n\,\mathrm{d}V \tag{94}$$

The volume integral in Eq. (94) leads to:

$$\tilde{P}_j^n = p_j^n \int_0^{\Delta x} \mathrm{d}x \int_0^{\Delta t/2} \mathrm{d}t = \frac{\Delta x \Delta t}{2} p_j^n \tag{95}$$

The numerical analogue of Eq. (82) becomes:

$$\tilde{F}_j^n = \tilde{P}_j^n \tag{96}$$

With the aid of Eqs. (92) and (95), Eq. (96) implies that:

$$\left[2u_j^n - \Delta t p_j^n\right] = \left[u_{j+1/2}^{n-1/2} + u_{j-1/2}^{n-1/2} - s_{j+1/2}^{n-1/2} + s_{j-1/2}^{n-1/2}\right] \tag{97}$$

Equation (97) is a nonlinear algebraic equation in terms of u_j^n, which originates from a nonlinear source term (p_j^n). Since this system of equations should be solved iteratively (e.g., using a Newton's iteration method), it is called the iterative CE/SE method, where Jacobian matrices f_u and p_u are required in Eq. (90) and (97).

Here, Δx and Δt are user-supplied parameters. How their values should be chosen is problem-dependent. A small spatial step size (Δx) should be chosen for a problem associated with steep moving fronts. Also, a small CFL number $\left(v = \frac{a\Delta t}{\Delta x}\right)$ is preferred for a problem that is stiff with respect to time. Note that the stability of a CE/SE scheme requires that the CFL number $|\nu| < 1$ (Chang 1995), as mentioned for explicit schemes in Section 2.4.1.

Without using special techniques that involve ad hoc parameters, the numerical dissipation associated with a CE/SE simulation with a fixed total marching time generally increases as the CFL number decreases. As a result, for a small CFL number (say $|\nu| < 0.1$), a CE/SE scheme may become overly dissipative. To overcome this shortcoming, a new CFL number insensitive scheme, i.e., the so-called Scheme II, was introduced in Chang (2002). The new scheme differs from other CE/SE schemes only in how $(u_x)_j^n$ is evaluated. Refer to Chang (2002) or Lim et al. (2004) for the detailed formulation.

2.4.3.2 Noniterative CE/SE Method

The noniterative CE/SE method is simply obtained from a first-order Taylor approximation of the source term (Molls and Molls 1998; Lim and Jørgensen 2004).

$$\tilde{p}_j^n = p_j^n + p_{x_j}^n(x - x_j) + p_{t_j}^n(t - t^n) \tag{98}$$

Using the above equation, Eq. (95) is replaced by

$$\begin{aligned}\left(\tilde{P}_\pm\right)_j^n &\equiv \iint_{CE\pm} \tilde{p}_j^n \, \mathrm{d}t\,\mathrm{d}x = \int_0^{\Delta x/2}\int_0^{\Delta t/2}\left(p_{j\pm1/2}^{n-1/2} + p_{x_{j\pm1/2}}^{n-1/2}\cdot x + p_{t_{j\pm1/2}}^{n-1/2}\cdot t\right)\mathrm{d}t\,\mathrm{d}x \\ &= \frac{\Delta x}{2}\frac{\Delta t}{8}\left(4p_{j\pm1/2}^{n-1/2} + \Delta x p_{x_{j\pm1/2}}^{n-1/2} + \Delta t p_{t_{j\pm1/2}}^{n-1/2}\right)\end{aligned} \tag{99}$$

where the time and space derivatives of source terms (p) are reformulated through the chain rule:

$$p_t \equiv \frac{\partial p}{\partial u}\frac{\partial u}{\partial t} = p_u u_t \tag{100a}$$

$$p_x \equiv \frac{\partial p}{\partial u}\frac{\partial u}{\partial x} = p_u u_x \tag{100b}$$

With the aid of Eq. (99), Eq. (97) evaluated on CE(j, n) is replaced by:

$$u_j^n = \frac{1}{2}\left[u_{j+1/2}^{n-1/2} + u_{j-1/2}^{n-1/2} - s_{j+1/2}^{n_1/2} + s_{j-1/2}^{n-1/2} + w_{j+1/2}^{n-1/2} + w_{j-1/2}^{n-1/2}\right] \tag{101}$$

where $w_{j\pm1/2}^{n-1/2} = \frac{\Delta t}{8}$ $(4\,p_{j\pm1/2}^{n-1/2} + \Delta x p_{xj\pm1/2}^{n-1/2} + \Delta t p_{tj\pm1/2}^{n-1/2})$. $(u_x)_j^n$ is also evaluated by Scheme II proposed by Chang (2002). Thus, two unknowns (u_j^n, u_{xj}^n) are obtained from four known values $(u_{j\pm1/2}^{n-1/2}, u_{xj\pm1/2}^{n-1/2})$ at the previous time level $(t^{n-1/2})$. u_j^n in Eq. (101) is obtained without nonlinear iteration procedure. This scheme is a noniterative CE/SE scheme, where Jacobian matrices f_u and p_u evaluated at the previous time level ($t = t^{n-1/2}$) are required in Eq. (101).

2.4.3.3 Boundary Conditions

Boundary conditions (at j = *1* and N_{mesh}) for state variables (u) and its spatial derivatives (u_x) are needed only at each integer-time level (n = 0, 1, 2, 3, ...) because of the staggering mesh structure and the intrinsically space-time triangle computational elements (see Fig. 2.7). At each half-time level (n = 1/2, 1 + 1/2, 2 + 1/2, ...), the values of u and u_x for all mesh points (j = 1 + 1/2, 2 + 1/2, ..., N_{mesh}–1/2) are calculated on the basis of the values at the previous integer-time level without requiring boundary values.

When the Danckwert boundary condition Eq. (3) is applied, conservative boundary conditions (BCs) at $x = x_0$ and $x = x_f$ can be constructed within $CE_+(l, n)$ and $CE_-($-$N_{mesh}, n)$, respectively. Performing a line integral along $CE_+(l, n)$ and using Eq. (3a), the boundary condition at $x = x_0$ (i.e., j = 1) for the iterative CE/SE method is obtained:

$$u_j^n - \frac{\Delta t}{2} p_j^n + q_j^n = u_{j+1/2}^{n-1/2} - s_{j+1/2}^{n-1/2} \tag{102a}$$

$$f_j^n = f_{in}^n + D \cdot u_{x_j}^n \tag{102b}$$

where $q_j^n = \frac{\Delta x}{4} u_{xj}^n - \frac{\Delta t}{\Delta x} f_j^n + \frac{\Delta t^2}{4\,\Delta x} f_{tj}^n$ and f_{in}^n is the inlet flux predefined by the operation condition. Eq. (102) leads to a nonlinear equation with respect to two variables, $u_j^n$and $u_{xj}^{n+1}$when j = 1.

The boundary condition at $x = x_f$ (i.e., $j = N_{mesh}$) for the iterative CE/SE method is obtained in the same way but by performing a line integral along $CE_-(N_{mesh}, n)$:

$$u_j^n - \frac{\Delta t}{2} p_j^n + \frac{\Delta t}{\Delta x} f_j^n = u_{j-1/2}^{n-1/2} + s_{j-1/2}^{n1/2} \tag{103a}$$

$$u_{x_j}^n = 0 \tag{103b}$$

Here, since q_j^n can reduce to $q_j^n = -\frac{\Delta t}{\Delta x} f_j^n$ with the aid of Eq. (90) and (103b), u_j^n is computed from Eq. (103a) through a nonlinear iteration.

For the noniterative CE/SE simulation, Eq. (102a) and (103a) are replaced, respectively, by:

$$u_1^n + q_1^n = u_{1+1/2}^{n-1/2} - s_{1+1/2}^{n-1/2} + w_{1+1/2}^{n-1/2} \tag{104}$$

$$u_{N_{\text{mesh}}}^n + \frac{\Delta t}{\Delta x} f_{N_{\text{mesh}}}^n = u_{N_{\text{mesh}}}^{n-1/2} + s_{N_{\text{mesh}}}^{n-1/2} + w_{N_{\text{mesh}}}^{n-1/2} \tag{105}$$

When other boundary conditions are imposed, appropriate formulations can be derived in a conservative manner within the conservation elements ($CE_\pm$ (j, n)).

2.4.3.4
Comparison of CE/SE Method with Other Methods

The iterative CE/SE method, at each time level, is associated with a block diagonal Jacobian matrix. Let the number of PDEs and spatial mesh points be N_{PDE} and N_{mesh}, respectively. The maximum number of nonzero Jacobian elements for the CE/SE method, $J_{max}^{CE/SE}$, is:

$$J_{\max}^{\mathrm{CE/SE}} = (N_{\mathrm{PDE}} \times N_{\mathrm{PDE}}) \times N_{\mathrm{mesh}} \tag{106}$$

In the case of linear source terms or noniterative CE/SE simulations, the Jacobian matrix is further reduced to a diagonal form:

$$J_{\min}^{\mathrm{CE/SE}} = (N_{\mathrm{PDE}} \times 1) \times N_{\mathrm{mesh}} \tag{107}$$

When an implicit ODE integrator is used in the MOL framework for Eq. (78), a band matrix is obtained. Let the length of the upper and lower band matrix be MU and ML dependent on the spatial discretization and nonlinearity of the PDE considered. The maximum number of nonzero band-Jacobian elements for the MOL is known as (Lim et al. 2004):

$$J_{\max}^{\mathrm{MOL}} = N_{\mathrm{PDE}} \cdot N_{\mathrm{mesh}}(ML + MU + 1) - \frac{1}{2}ML(ML+1) - \frac{1}{2}MU(MU+1) \tag{108}$$

For example, in the simple case where the convection term is discretized by a first-order backward scheme and the diffusion term by a central scheme like Eq. (9), $ML = MU = N_{PDE}$. The smallest number of nonzero Jacobian elements in this case, J_{min}^{MOL}, can be approximated at each time step:

$$J_{\min}^{\mathrm{MOL}} \approx (2N_{\mathrm{PDE}} \times N_{\mathrm{PDE}}) \times N_{\mathrm{mesh}} \tag{109}$$

As a result, the following relation can be derived:

$$J_{\min}^{\mathrm{CE/SE}} < J_{\max}^{\mathrm{CE/SE}} \leq J_{\min}^{\mathrm{MOL}} \leq J_{\max}^{\mathrm{MOL}} \tag{110}$$

Eq. (110) means that the number of nonzero Jacobian elements for the MOL, J_{max}^{MOL}, is not less than $J_{max}^{CE/SE}$. The computational time is normally proportional to the number of nonzero Jacobian elements (J) multiplied by the number of time steps (N_{time}), i.e., $J \times N_{time}$. Therefore, it is expected that the computational time of the CE/SE method is shorter than the MOL for the same number of time steps. Especially for nonstiff systems (e.g., chromatographic adsorption problems), the CE/SE method will save computational time because a small number of time steps can be used (Lim et al. 2004). In Section 2.6, the MOL and the CE/SE methods are compared for several PDE problems in terms of accuracy and computational efficiency.

In Chang et al. (2000), the CE/SE method is compared with the Leapfrog, Lax-Wendroff, DuFort-Frankel and MacCormarck schemes (see Section 2.4.1). Here, the CE/SE method shows promising performance compared to these fully discrete methods.

While the implicit ODE integrator has a self-adaptive feature, i.e., variable order and time stepsize (Δt), the present CE/SE method has a fixed value of Δt satisfying the CFL condition. Thus, for stiff problems a main disadvantage of the CE/SE method could be the fixed time step (Δt).

2.5 Advanced Numerical Methods

Adaptive mesh methods can improve the accuracy and efficiency of the numerical approximations to evolutionary PDE systems that involve large gradients or discontinuities. As the solution changes in an evolutionary PDE, the mesh must also change to adaptively refine regions where the solution is developing sharp gradients, and to remove points from regions where the solution is becoming smoother (Li 1998).

Over the past years, significant interest has been devoted to adaptive mesh methods. Various sophisticated techniques have been proposed. For example, adaptive mesh refinement (AMR) removes/adds the nodes at discrete time levels and moving grid methods function by moving the nodes continuously over time (Vande Wouwer et al. 1998).

Adaptive mesh methods have important applications for a variety of physical and engineering problems (e.g., solid/fluid dynamics, combustion, heat transfer, etc.) that require extremely fine meshes in a small part of the physical domain. Successful implementation of the adaptive strategy can increase the accuracy of the numerical approximation and also decrease the computational cost. This section addresses two different strategies: AMR and the moving mesh method.

2.5.1 Adaptive Mesh Refinement

The AMR approach (Berger and Oliger 1984; Berger and LeVeque 1998) has been shown to be one of the most effective adaptive strategies for PDEs and refines in space and/or time. The AMR process is composed of three steps: error estimation, mesh refinement and solution interpolation. An AMR package called the conservation laws package (CLAWPACK) from the University of Washington is available from http://www.amath.washington.edu/~claw/.

2.5.1.1 Error Estimation

One way to estimate errors is to use a weighted combination of first and second solution differences. The error (E_i) at $x = x_i$ is estimated to be:

$$E_i = \sum_{k=1}^{N_{\mathrm{PDE}}} w_1 \left|u_{k,i+1} - u_{k,i}\right| + w_2 \left|u_{k,i+1} - 2u_{k,i} + u_{k,i-1}\right|, \quad i = 2 \ldots (N_{\mathrm{mesh}} - 1) \quad (111)$$

where N_{PDE} denotes the number of PDEs and the weighting factors w_1 and w_2 are user-defined. In the adaptive algorithm, the mesh is refined in portions of the physical domain where the inequality

$$E_i \geq \varepsilon \tag{112}$$

is satisfied and where ε is a user-specified error tolerance.

2.5.1.2 Mesh Refinement

AMR adds new refinement grids where the error is estimated to be large. The refinement grids are usually aligned with the underlying base grid. The refinement grids are arranged in a hierarchy, with the base grids belonging to level one, the next grids being added to level 2 and so on. Grids on level m are refined by a refinement ratio r, (usually 2 or 4) from the grids on level $(m - 1)$. The grids are normally properly nested so that a grid on level m is completely contained in the grids on level $(m - 1)$. A hierarchical block grid structure for two-dimensional AMR with $r = 2$ is shown in

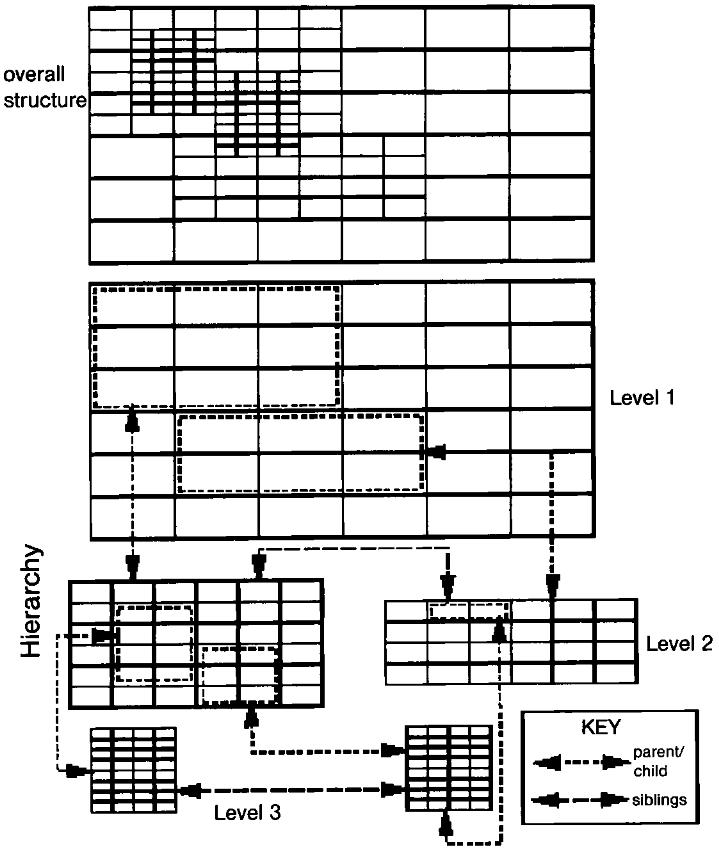


Figure 2.8 A hierarchical block grid structure of AMR

Fig. 2.8 (Li 1998). Each refinement level consists of several blocks. A block is a logically rectangular grid. After each refinement, the refined cells are clustered into several blocks. Buffer zones and ghost boundaries may be added to each block. All the blocks are managed by a hierarchical data structure.

The data structure in the AMR algorithm is complex due to the existence of several levels (m). If $u(x_i, t^n, m)$ were used to store the data, it would be a waste of memory, because at higher refined levels, only a small part of the grid is used. In order to efficiently manage all the discrete points at the same level, we need to cluster them into several disconnected segments, called patches, and treat the patch as the basic data unit (Li 1998). The patches are building blocks of the hierarchical grid structure.

2.5.1.3 Solution Interpolation

The regridding includes computing the physical locations for each fine grid and copying or injecting the solution from the old grid to the new grid. The physical mesh positions are easy to compute by linear interpolation. The solution needs more attention. Although the solution can be obtained from the coarse grid by injection or interpolation, a more accurate solution is obtained from the old grid at the same level, which partially overlays the new grid.

One of the secrets behind the success of the AMR algorithm is that flow discontinuities always fall within the overlay regions between the new and old grid. Thus the adaptation process cannot introduce further errors in these problem regions by solution interpolation. The method used to interpolate the solution from the coarse grid needs to be chosen with care.

Conservative interpolation is useful in regions near a discontinuity. The coarse grid solution is assumed to be piecewise linear. The slopes for each grid are found by applying a MinMod limiter function to the forward and backward slopes between cell centers (Li 1998). So, for a coarse cell i,

$$\begin{aligned} &u_{i+1/2} - u_{i-1/2} = \mathrm{MinMod}(u_{i+1} - u_i,\ u_i - u_{i-1}) \\ &\text{where } \mathrm{MinMod}(a, b) = \begin{cases} 0, & \text{if } ab < 0 \\ \mathrm{sign}(a) \cdot \min(|a|, |b|), & \text{elsewhere} \end{cases} \end{aligned} \tag{113}$$

2.5.1.4 Boundary Conditions

There are two types of boundaries in an AMR system: external boundaries and internal boundaries. External boundaries are given by the problem definition and internal boundaries are generated by refinement. Each patch in one-dimension has two ends: the left and the right. The boundary values are often collected only at the backward time t^{n-1} just before the integration from t^{n-1} to t^n. This causes a problem when the time integration is performed by an implicit or higher-order MOL, because the boundary values at t^n are usually required to compute the intermediate time derivatives of the boundary cells in an MOL approach. This problem can be solved by collecting the values for the internal boundaries from the parent coarse grid at the forward time t^n before integrating the current time level.

2.5.2 Moving Mesh Methods

Although AMR or local mesh refinement is quite reliable and robust, Furzeland et al. (1990) stated that it is cumbersome in some cases to apply it because of the interpolation procedure, nonfixed number of grid points, restart of integration at certain time steps, etc. The moving grid methods (Miller and Miller 1981; Dorfi and Drury 1987; Huang and Russell 1997), where the number of mesh points is kept unchanged, could be very powerful due to the continuous grid adaptation with the evolution of the solution.

In the MOL framework, Furzeland et al. (1990) consider the moving finite difference (MFD) approach (Dorfi and Drury 1987; Huang and Russell 1997) as promising with respect to reliability, efficiency and robustness. The moving finite element (MFE) approach (Miller and Miller 1981; Kaczmarski et al. 1997; Liu and Jacobsen 2004) that enables one to handle more complicated physical domains may be considered difficult to use because of tuning parameters and to be computationally inefficient.

Moving mesh methods have traditionally used a finite difference method (normally with a simple three point central difference) to discretize both the physical PDE and moving mesh PDE (MMPDE). The MFD approach using the central discretization proposed by Huang and Russell (1997) and Dorfi and Drury (1987) is still unstable in some cases because of the central discretization of first-order derivatives.

The ENO and WENO methods (Shu and Osher 1989; Jiang and Shu 1996) are uniformly high-order accurate right up to the discontinuity. Moreover these methods may well be applied to the moving grid method due to the reliable numerical results of first-order derivatives. Li and Petzold (1997) presented a combination of the moving grid method of Dorfi and Drury (1987) with the ENO schemes (Shu and Osher 1989) in order to improve stability and accuracy in the discretization procedure.

We are interested in the numerical solution of well-posed systems of PDEs, e.g., in Eq. (1a). If meshes are moving continuously with time, i.e, $x_i = x_i(t)$, by the chain rule, the solution of Eq. (1a) satisfies the following equation (Dorfi and Drury 1987):

$$\begin{cases} \dot{u} = -f_x + Du_{xx} + r(u) + u_x\dot{x} \\ 0 = m(u, x, \dot{x}) \end{cases} \tag{114}$$

where $\dot{u}$ and $\dot{x}$ denote the time derivatives of u and x, respectively, when nonuniform physical meshes (x_i) are transformed into uniform computational meshes (ξ_i). Mesh movement is governed by $m(u, x, \dot{x})$. The PDE and the mesh equation are intrinsically coupled and are generally solved simultaneously.

2.5.2.1 Equidistribution Principle

The grid equation, $m(u, x, \dot{x})$, is induced from the equidistribution principle (EP), which means that the grids are spaced in order to make each arc length of discrete solutions equally distributed at each grid step. Therefore, the nodes are concentrated

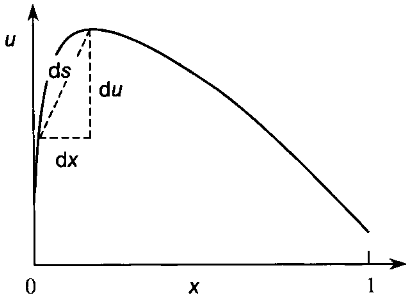


Figure 2.9 Arc length ($ds = \sqrt{ds^2 + du^2}$) of the solution at a time level (t)

in steep regions. The one-dimensional EP can be expressed in its integral form with the computational coordinate ($0 \leq \xi \leq 1$) and the monitor function, M(x, t), as a metric of each arc length (see Fig. 2.9):

$$\int_0^{x_i(\xi,t)} M(x,t)\,\mathrm{d}\xi = \frac{i}{N_{\mathrm{mesh}}}\int_0^1 M(x,t)\,\mathrm{d}\xi, \quad i = 1\ldots N_{\mathrm{mesh}} \tag{115}$$

where M(x, t) is called the monitor function. For example, as shown in Fig. 2.9, the arc length monitor function is given:

$$M(x,t) = \frac{\mathrm{d}s}{\mathrm{d}x} = \sqrt{1 + (u_x)^2} \tag{116}$$

Let the total arc length of a numerical solution at a time t be $\theta(t($ = $\int_0^1 M\ (x,\ t)\ d\xi$. Therefore, Eq. (115) is replaced by:

$$\int_0^{x_i(\xi,t)} M(x,t)\,\mathrm{d}\xi = \xi_i\theta(t) \tag{117}$$

where $\xi_i = i/N_{mesh}$ is the uniform computational coordinate mentioned above. $\theta(t)$ is fixed at a given time regardless of ξ and is unknown. As it is difficult to treat this unknown term, $\theta(t)$, in the numerical procedure, it is eliminated by differentiating Eq. (117) with respect to ξ once and twice. A quasistatic EP is so obtained:

$$\frac{\partial}{\partial \xi}\left\{M(x(\xi,t),t)\frac{\partial x(\xi,t)}{\partial \xi}\right\} = 0 \tag{118}$$

From equation (118) one can obtain various mesh equations involving node speeds ($\dot{x}$), the so-called moving mesh PDE (MMPDE), which are employed to move a mesh having a fixed number of nodes in such a way that the nodes remain concentrated in regions of rapid variation of the solution (Huang and Russell 1997).

For most discretization methods of Eq. (114), abrupt variations in the mesh will cause deterioration in the convergence rate and an increase in the error (Huang and Russell 1997). Moreover, most discrete approximations of spatial differential operators (e.g., u_x in Eq. (114)) have much larger CFL condition numbers (e.g., $= \left(\frac{\partial f}{\partial u} + \dot{x}\right)\frac{\Delta t}{\Delta x_i}$ in Eq. (114)) on an abruptly varying mesh than they do on a gradually varying one. The ill-conditioned approximations may result in stiffness in the time integration in the framework of MOL. Robust mesh equations spatially smoothed are proposed by Dorfi and Drury (1987) and Huang and Russell (1997).

As an illustration, the well-known Burgers' equation with a smooth initial condition is considered:

$$u_t = -uu_x + 10^{-4}u_{xx}, \quad 0 \le x \le 1 \tag{119}$$

$$u(x, 0) = \sin(2\pi x) + \sin(\pi x)/2 \tag{120}$$

The boundary condition is given as $u(0, t) = 0.0$ and $u(1, t) = 0.0$. A uniform grid structure is used as the initial grid position. The solution is a wave that develops a very steep gradient and subsequently moves towards $x = 1$. Because of the zero boundary values, the wave amplitude diminishes with increasing time. This is quite a challenging problem for both fixed and moving mesh methods. Proper placement of the fine mesh is critical, and a moving grid method tends to generate spurious oscillation as soon as the mesh becomes slightly too coarse in the layer region, just like nonmoving mesh methods with a central difference (Lim et al. 2001b).

Figure 2.10 shows numerical results of the Burgers' equation solved by the MOL with the third-order WENO scheme on 40 moving grid points (Lim et al. 2001b). The mesh points are well concentrated on the moving front and adapt to physical fluid flow. In Fig. 2.11, grid evolution with time is shown for this case. The mesh points move continuously according to variation of the solution. The moving grid method attains a resolution corresponding to 5000 equidistant grid points near the shock, and to 7.5 equidistant grid points near the smooth regions.

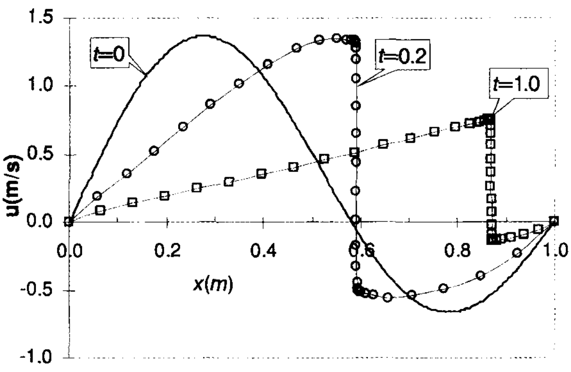


Figure 2.10 Numerical solutions of Burgers' equation on 40 moving grid points

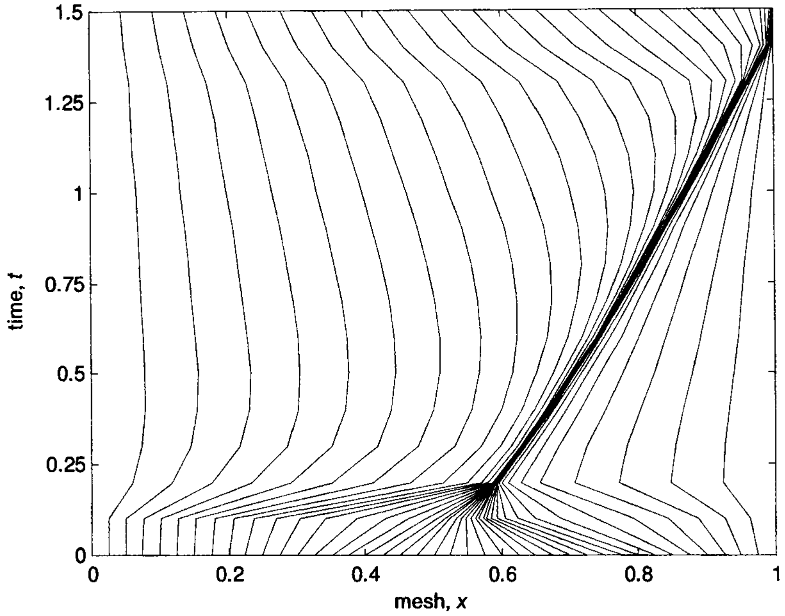


Figure 2.11 Grid evolution with time on 40 moving grid points

2.5.3 Comparison between AMR and Moving Mesh Method

In the numerical analysis of PDAEs, discretization methods on fixed mesh points are generally more robust and easy-to-use than those on adaptive mesh points. In the cases involving steep moving fronts, the adaptive mesh methods are efficient with respect to accuracy and computational time. AMR and moving mesh methods are two of the most successful adaptive mesh methods. However, some care is needed to successfully use them. AMR has been developed for explicit temporal integration, while the moving mesh method works efficiently for implicit temporal integration (e.g., MOL in Section 2.3) because of stiffness of the grid.

Moving-grid methods use a fixed number of spatial grid points, without need for interpolation. Moving mesh methods implemented via implicit time integration take advantage of the fully automatic adaptation of temporal and spatial stepsize (Δt_n and Δx_i). However, simultaneous solution procedures of physical and mesh equations typically suffer from the large computation time due to highly nonlinear coupling between the two equations, often requiring an excessive Newton iteration at each time step. This problem is further exacerbated by the dense clustering of mesh points near discontinuities, which degrades the convergence of the iteration (Stockie et al. 2001). Moreover, the extension of moving mesh methods from one dimension to higher dimensions in not straightforward (Li 1998). The two-dimensional moving mesh equation is much more complicated, because it includes many factors such as temporal smoothness, orthogonality and skewness of the mesh (Huang and Russell

1999). Simplicity and efficiency for the extension from 1D to 2D/3D motivate development of the local refinement method such as AMR (Li 1998). Data structure and algorithms in AMR for a one-dimensional grid can be extended to higher dimensions without difficulty.

Adaptive mesh methods also introduce overhead. For the moving mesh, such overhead includes evaluation of the monitor function, regularization of the mesh function, computation of the mesh velocity and solving additional mesh equation for the node positions. Compared with the moving mesh method, the overhead for the AMR method is much less. The evaluation of the monitor function is much cheaper and there is no need for regularization of the mesh function. Most of the overhead comes from the refinement and management of the hierarchical data structure (Li 1998).

Recently, the combination of the two adaptive mesh strategies was presented (Hyman et al. 2003) and the two methods are compared and reviewed.

2.6 Applications

This section illustrates applications of the introduced numerical methods for the solution of PDE or PDAE systems in several dynamic chemical/biochemical processes. In Table 2.6, the five examples to be presented are characterized according to the type of equations and physical dominant phenomena. Each of the five problems is described by a time-dependent process model within one-dimensional space.

First, chromatography columns modeled by a PDAE system are presented in Section 2.6.1. Here, numerical performances of several MOL methods and the CE/SE method are compared for both linear and nonequilibrium adsorption. In the fixed-

Table 2.6 Classification of application examples

Section		Type of equations	Physical meanings related	Characteristics
2.6.1	Chromatography	PDAE with source term	Convection, diffusion, and adsorption	Steep moving fronts
2.6.2	Fixed-bed reactor	PDE with source term and recycle	Convection, diffusion, and reaction	Mass and heat recycle and oscillation profiles
2.6.3	Slurry bubble column reactor	PDE with source term	Convection, diffusion, and reaction	Chemical reaction related to three-phase hydrodynamics
2.6.4	Population balance equation	Integro-PDE with source terms	Growth, nucleation, agglomeration, and breakage	Dynamic behaviors of the particle size with discontinuous fronts
2.6.5	Cell population dynamics	Integro-PDE with source terms	Growth and cell division	Oscillatory behaviors of cell populations

bed reactor model (see Section 2.6.2), oscillatory behaviors of state variables caused by mass/energy recycling are examined and several numerical methods are compared. In Section 2.6.3, a slurry bubble column reactor for Fischer-Tropsch synthesis is considered, where three-phase hydrodynamics are modeled by empirical equations given by De Swart and Krishna. (2002). The dynamics of gas/liquid concentrations and temperature are predicted at the beginning of operation. A population balance equation modeling crystal growth, nucleation, agglomeration and breakage is solved in Section 2.6.4, where discontinuous moving fronts appear due to initial seed crystals. Finally, Section 2.6.5 considers cell population dynamics in microbial cultures described by cell population balance equation (PBE) coupled to metabolic reactions relevant to extracellular environment (Zhu et al. 2000).

2.6.1 Chromatography

Packed-bed chromatographic adsorption between the stationary and mobile phases leads, for each component, to a partial differential algebraic equation (PDAE) system involving one partial differential equation (PDE), one ordinary differential equation (ODE) and one nonlinear algebraic equation (AE) (Lim et al. 2004):

$$\frac{\partial C}{\partial t} = -\frac{\partial (v_L C)}{\partial z} + \frac{\partial}{\partial z}\left(D_{ax}\frac{\partial C}{\partial z}\right) - \alpha\frac{dn}{dt} \tag{121a}$$

$$\frac{dn}{dt} = k(n^* - n) \tag{121b}$$

$$0 = g(C, n^*) \tag{121c}$$

where v_L is the interstitial velocity, D_{ax} is the axial dispersion coefficient, α is the volume ratio between the two phases, and k refers to the mass transfer coefficient. The liquid and solid concentrations for each component are referred to as C and n, respectively. n^* is the equilibrium concentration (or adsorption isotherm). Since the Peclet number (ratio of convection to diffusion, $Pe = \frac{v_L L_c}{D_{ax}}$, *where* L_c is the column length) is often large in chromatographic processes (Poulain and Finlayson 1993), Eq. (121) is classified as a convection-dominated parabolic PDAE system.

The packed-bed chromatographic problem in Eq. (121), is solved for one component with the volume ratio $\alpha = 1.5$, the fluid velocity $v_L = 0.1$ m/s, the axial dispersion coefficient $D_{ax} = 1.0 \times 10^{-5}$ m^2/s, and the adsorption rate coefficient $k = 0.0129$ s^{-1}. A linear adsorption isotherm is used for the algebraic Eq. (121c).

$$n^* = 0.85C \tag{122}$$

The column length is in the interval $0 \leq z \leq 1.5$ and the integration time is $0 \leq t \leq 10$ s. As the initial condition, $C(0, z) = 0$, $n(0, z) = 0$ and $n^*(0, z) = 0$ for all z except $z = 0$ and $z = 1.5$.

Suppose that the Danckwert's boundary condition for Eq. (121a) is imposed as below:

$$\text{At } z = 0 \text{ and } \forall t, \quad v_L(C - C_{\text{in}}) = D_{ax} \cdot \frac{\partial C}{\partial z} \tag{123a}$$

$$\text{At } z = 1.5 \text{ and } \forall t, \quad \frac{\partial C}{\partial z} = 0 \tag{123b}$$

where C_{in} is a known feed concentration just before entering to the column. Here, an inlet square concentration pulse is considered as follows:

$$C_{\text{in}} = 2.2, \ \text{for } 0 \le t \le 2.0s \tag{124a}$$

$$C_{\text{in}} = 0.0, \ \text{for } 2.0s \le t \le 10.0s \tag{124b}$$

The numerical solutions are obtained on 201 equidistant spatial mesh points. The CFL number for the iterative CE/SE method (Lim et al. 2004; see also Section 2.4.3) is given at $v = 0.4$. The reference solution is obtained on 401 equidistant mesh points through the iterative CE/SE method. The error is estimated using Eq. (36).

Table 2.7 reports numerical performance on accuracy, computational efficiency and stability for the chromatographic adsorption problem with axial dispersion on 201 mesh points. The second-order central and fifth-order upwinding schemes give spurious oscillatory solutions near steep regions. Thus, the two methods seem to be inadequate for convection-dominated problems as mentioned in Lim et al. (2001a). The first-order upwinding scheme (called first-order upwind, or FS-upwind-1) is not accurate because of its low order of accuracy. The two WENO schemes (third-order and fifth-order) enhance accuracy and stability but at the cost of longer computation time. The CE/SE method gives, in this case study, the most accurate solution with very short calculation times in a stable manner.

In Fig. 2.12 numerical solutions of the fluid concentration (C) are depicted near $z = 0.9$ at $t = 10$ s for the adsorption problem. The reference solution is a smeared square profile at $z = 0.8$ and 1. The CE/SE method shows the best solution without

Table 2.7 Accuracy, temporal performance and stability evaluation for a chromatographic adsorption PDAE with axial dispersion and square input concentration on 201 mesh points

		Accuracy (L1 error)**	CPU time (s)***
MOL	FS-upwind-1	0.2075	1.6
	FS-central-2	0.0979*	1.9
	FS-upwind-5	0.0060*	2.9
	WS-upwind-3	0.0449	7.7
	WS-upwind-5	0.0168	11.3
Iterative CE/SE (CFL = 0.4)		0.0087	1.3

* Unstable numerical solution.

** L1 error at $t = 10$ s.

*** CPU time during 10-s integration time.

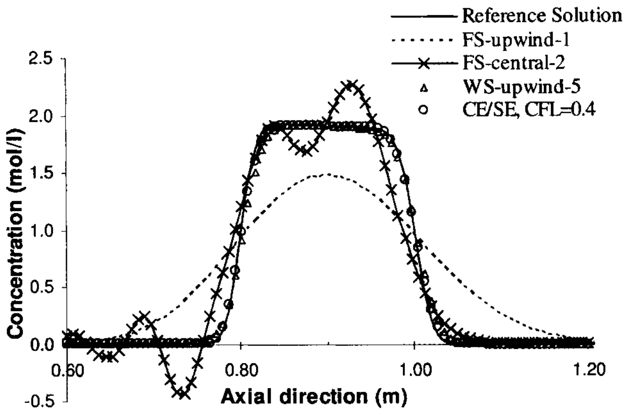


Figure 2.12 Fluid concentration (C) profiles for different numerical schemes around $z = 0.9$ at $t = 10$ s for the single component chromatographic adsorption problem with axial dispersion ($D_{ax} = 1.0 \times 10^{-5}$) and square input concentration on 201 mesh points

spurious oscillation of the six schemes tested. As expected, first-order upwind (or FS-upwind-1) is not accurate and second-order central (or FS-central-2) is highly oscillatory. The MOL with fifth-order WENO (or WS-upwind-5) and the CE/SE with $CFL = 0.4$ exhibit similar resolution in steep regions.

Figure 2.13 shows the propagation of steep waves with time. Note that the fifth-order WENO scheme (*circles*) and the CE/SE method (*solid line*) have a nondissipative

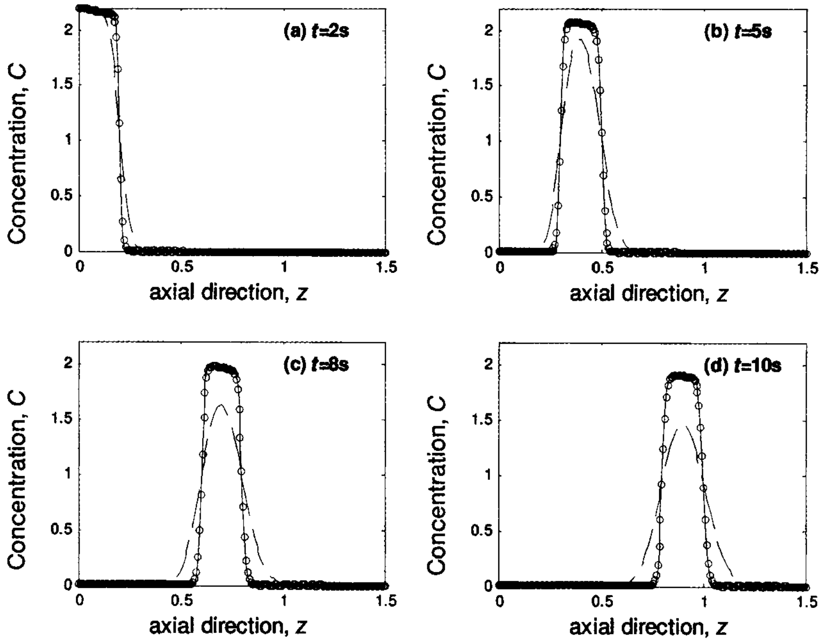


Figure 2.13 Fluid concentration (C) propagation with time for a chromatographic adsorption problem with axial dispersion on 201 mesh points (*dashed line*: first-order upwind; *circles*: fifth-order WENO; *solid lines*: CE/SE with $CFL = 0.4$)

feature owing to conservative discretization, since the waves do not widen with time. In contrast, the peak of the first-order upwinding scheme (*dashed line*) broadens continually as time increases.

2.6.2 Fixed-Bed Reactor

Recycling is often used in industrial processes to reduce the costs of raw materials and energy. The nonlinear effects of introducing recycling on a fixed-bed reactor are considered in plant-wide process control and bifurcation analysis (Recke and Jørgensen 1997). This nonlinearity has the most pronounced effect around bifurcation points, i.e., points where the system solutions change stability and/or number of possible solutions.

The fixed-bed reactor we consider here is a packed-bed tubular reactor with a single irreversible exothermic reaction of hydrogen's catalytic oxidation to form water (Hansen and Jørgensen 1976).

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \tag{125}$$

The reactor is mass- and heat-integrated, which means that unconverted reactants are recycled and the reactor effluent is used to preheat the reactor feed in an external heat-exchanger, as shown in Fig. 2.14.

The reaction is assumed to be first order in oxygen concentration with Arrhenius-type temperature dependence. The model describing the reactor with both mass and energy recycling is given by:

$$\kappa\frac{\partial y}{\partial t} = -v\frac{\partial y}{\partial \xi} + \frac{1}{Pe_m}\frac{\partial^2 y}{\partial \xi^2} - Da \cdot y \cdot e^{\gamma(1-1/\theta)} \tag{126}$$

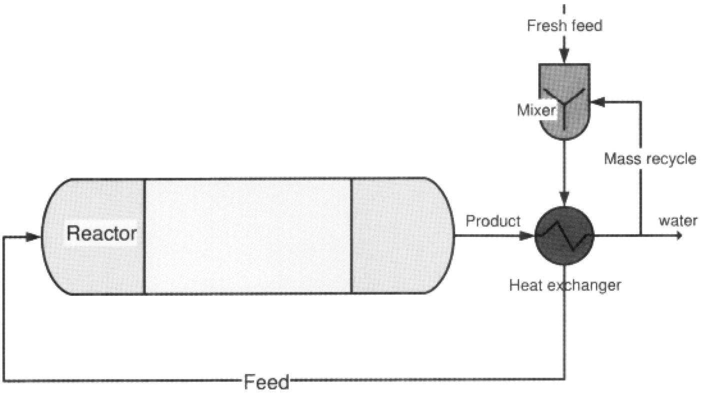


Figure 2.14 Schematic drawing of the fixed-bed reactor with mass/energy recycles

$$\frac{\partial\theta}{\partial t} = -v\frac{\partial\theta}{\partial\xi} + \frac{1}{Pe_H}\frac{\partial^2\theta}{\partial\xi^2} - Bi(\theta - \theta_c) + Be \cdot Da \cdot y \cdot e^{\gamma(1-1\theta)} \tag{127}$$

where the ratio of mass to thermal residence time $\varkappa = 1/600$, the dimensionless flow rate v) 1.0, the axial dispersion mass Peclet number $Pe_m = 270$, the Damkohler number $Da = 0.376$, the Arrhenius number $\gamma = 9.0$, the axial dispersion heat Peclet number $Pe_H = 118$, the Biot number $Bi = 0.5$, the dimensionless surrounding temperature $\theta_c = 0.79$, and the dimensionless heat reaction $Be = 0.49$ are used for simulation. The variables t, ξ, y and θ are the dimensionless time, axial direction, oxygen concentration and temperature, respectively. The mass and energy recycling are assumed to follow first-order dynamics:

$$\frac{\mathrm{d}y_{\mathrm{rec}}}{\mathrm{d}t} = \tau_m(y_{\xi=1} - y_{\mathrm{rec}}) \tag{128}$$

$$\frac{\mathrm{d}\theta_{\mathrm{rec}}}{\mathrm{d}t} = \tau_e(\theta_{\xi=1} - \theta_{\mathrm{rec}}) \tag{129}$$

where τ_m and τ_e denote mass and energy recycle time lag constants with the units $[t^{-1}]$, respectively. The above two ordinary differential equations have analytic solutions as follows:

$$y_{\mathrm{rec}} = y_{\xi=1} - (y_{\mathrm{rec},0} - y_{\theta=1,0})\,\mathrm{e}^{-\tau_m t} \tag{130}$$

$$\theta_{\mathrm{rec}} = \theta_{\xi=1} - (\theta_{\mathrm{rec},0} - \theta_{\xi=1,0})\,\mathrm{e}^{-\tau_e t} \tag{131}$$

where $y_{rec,()}$ and $y_{\xi=1,0}$ are the initial conditions for y_{rec}, and $y_{\xi=1}$ and $\theta_{\mathrm{rec},0}$ and $\theta_{\xi=1,0}$ are those for θ_{rec} and $_{\xi=1}$. Τηε ρεψυψλε τιμε λαγ ψονσταντσ $\tau_\mu = 30$ and τ_e are used in this simulation. The Danckwert boundary conditions at the inlet point ($\xi = 0$) are expressed as:

$$y_{\xi=0} = y_{\mathrm{feed}} + \alpha_m \cdot y_{\mathrm{rec}} + \frac{1}{Pe_m}\frac{\partial y}{\partial\xi}\bigg|_{\xi=0} \tag{132}$$

$$\theta_{\xi=0} = \theta_{\mathrm{feed}} + \alpha_e(\theta_{\mathrm{rec}} - \theta_{\mathrm{feed}}) + \frac{1}{Pe_H}\frac{\partial\theta}{\partial\xi}\bigg|_{\xi=0} \tag{133}$$

where the dimensionless feed oxygen concentration (y_{feed}) and temperature (θ_{feed}) are given as $y_{feed} = 1.0$ and $\theta_{\mathrm{feed}} = 0.8$ and the mass and energy recycle ratios are assumed to be α_m and α_e. The boundary conditions at the outlet point ($\xi = 1$) are given as:

$$\frac{\partial y}{\partial\xi}\bigg|_{\xi=1} = 0 \quad \text{and} \quad \frac{\partial\theta}{\partial\xi}\bigg|_{\xi=1} = 0 \tag{134}$$

The bed is initially set to no reactant (i.e., $y_0 = 0$ for all ξ at $t = 0$). The initial bed temperature is $\theta_0 = 0.79$ for all ξ.

The model is solved by the MOL and the noniterative CE/SE method for solution comparison. In the framework of the MOL, the convection term is discretized on uniform 201-mesh points by the first-order upwinding scheme (FS-upwind-1) and the third-order WENO scheme (WS-upwind-3), and the diffusion term by the central scheme (see Section 2.3.1). The boundary conditions Eqs. (132)–(134) are converted into nonlinear algebraic equations (AEs) by spatial discretization. The band Jacobian structure is broken by mass and energy recycles. The resulting system is thus a set of DAEs with a sparse Jacobian matrix.

Using the noniterative CE/SE method, the two coupled PDEs are fully discretized on uniform 201-mesh points at $CFL = 0.6$. The resulting system has 402 linear algebraic equations at each time level for $u_j^n$and u_{xj}^n, respectively.

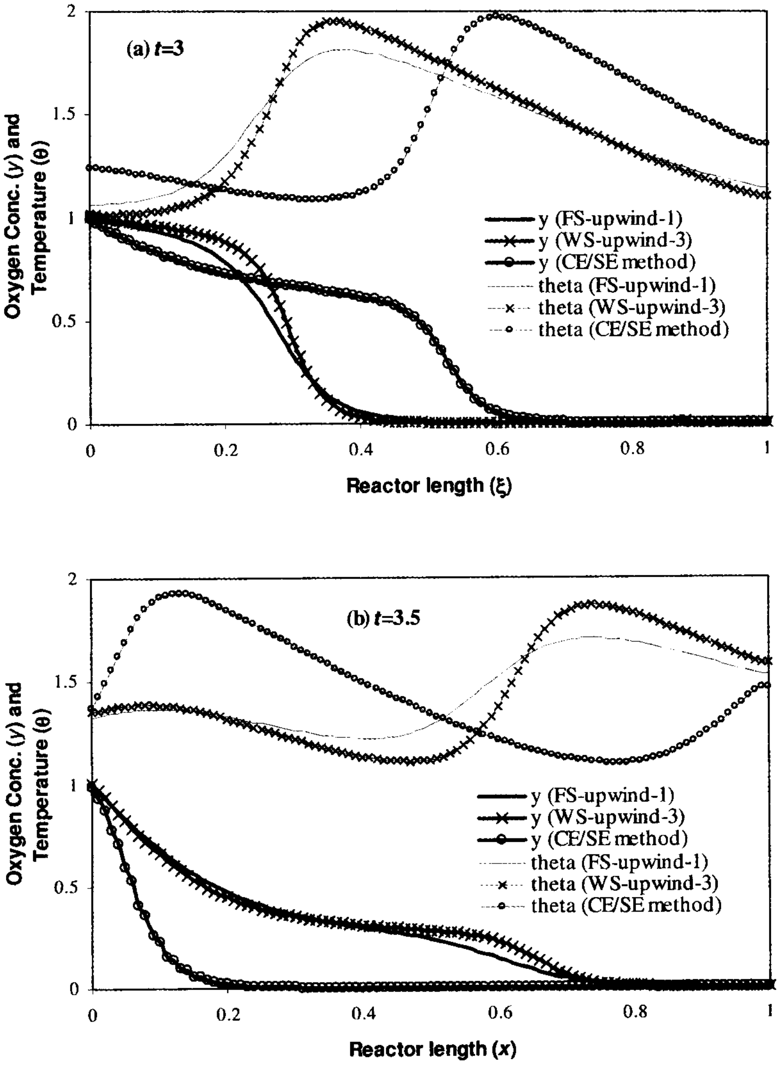


Figure 2.15 Comparison of numerical solutions for dimensionless oxygen concentration (y) and temperature (θ) variations with respect to the reactor length (x) at (a) $t = 3$ and (b) $t = 3.5$

Figure 2.15 shows the spatial distribution of dimensionless oxygen concentrations (y) and temperatures (θ) at two time levels, $t = 3$ and $t = 3.5$, for each of the three numerical methods. Even though smooth fronts move with time, the solution profiles depend highly on the numerical methods used due to mass/energy recycling. A different numerical method provides a different oscillatory frequency, phase degree, and/or amplitude. In Fig. 2.16, it is shown that a steady-state solution is reached differently depending on the numerical method used.

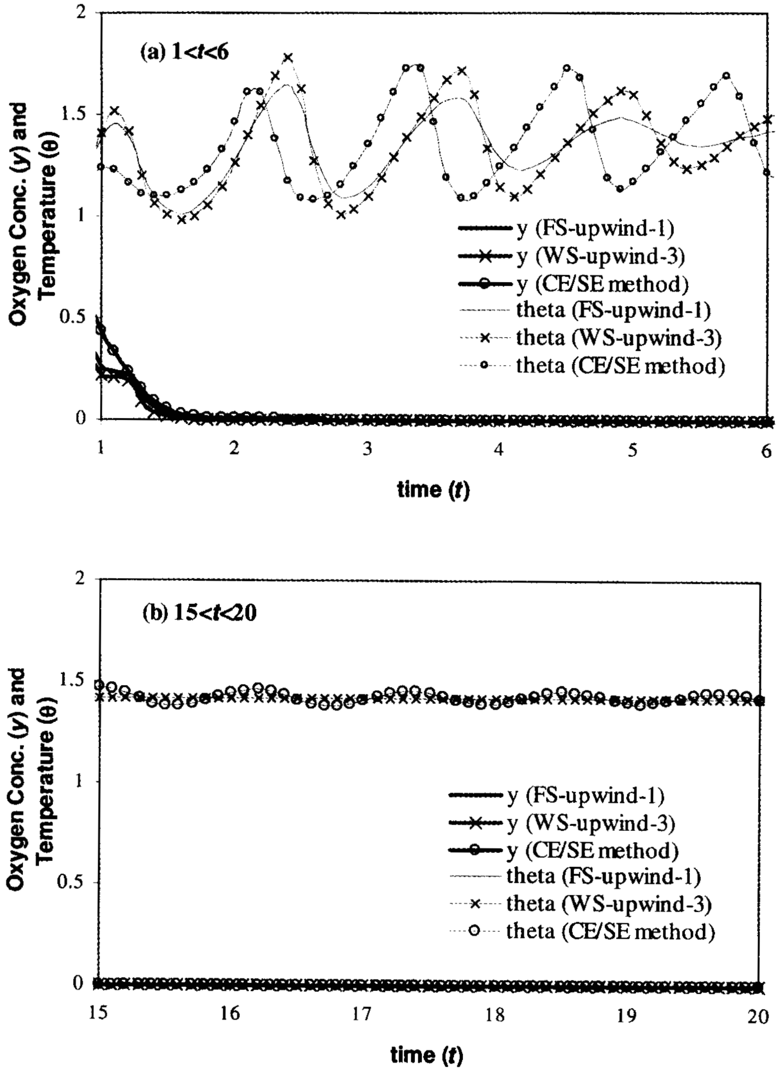


Figure 2.16 Comparison of numerical solutions for dimensionless oxygen concentration (y) and temperature (θ) variations with respect to time, (a) $1 < t < 6$ and (b) $15 < t < 20$, at the reactor outlet point ($\xi = 1$)

Liu and Jacobsen (2004) stated that some discretization methods such as finite differences and finite elements can result in spurious bifurcation and erroneous prediction of stability. To minimize discretization error, they proposed a moving mesh method (see Section 2.5.2), i.e., an orthogonal collocation method on moving finite elements, for solving a fixed-bed reactor model with energy recycling. For the fixed bed reactor system, there is clearly a need for checking the approximation error of spatial and temporal derivatives as the fronts move. This would mean for the CE/SE method that an adaptive mesh method is applied. In addition, it is questionable whether the exit boundary conditions Eq. (134) are physically reasonable, especially when steep fronts are moving out of the reactor.

2.6.3 Slurry Bubble Column Reactor

Slurry bed reactors are applied increasingly in the chemical industry. The specific example selected here focuses on Fischer-Tropsch (FT) synthesis. FT synthesis technology, such as fluidized bed, multitubular fixed-bed, and three-phase slurry bed, forms the heart of many natural gas conversion processes that have been developed by various companies in recent years (e.g., SASOL, Shell, Exxon, etc.). The FT reaction converts the synthesis gas (H_2+CO) into a mixture of mainly long straight chain paraffins. This example concerns the Fe-based (or Co-based) catalytic slurry bed reactor, as shown in Fig. 2.17.

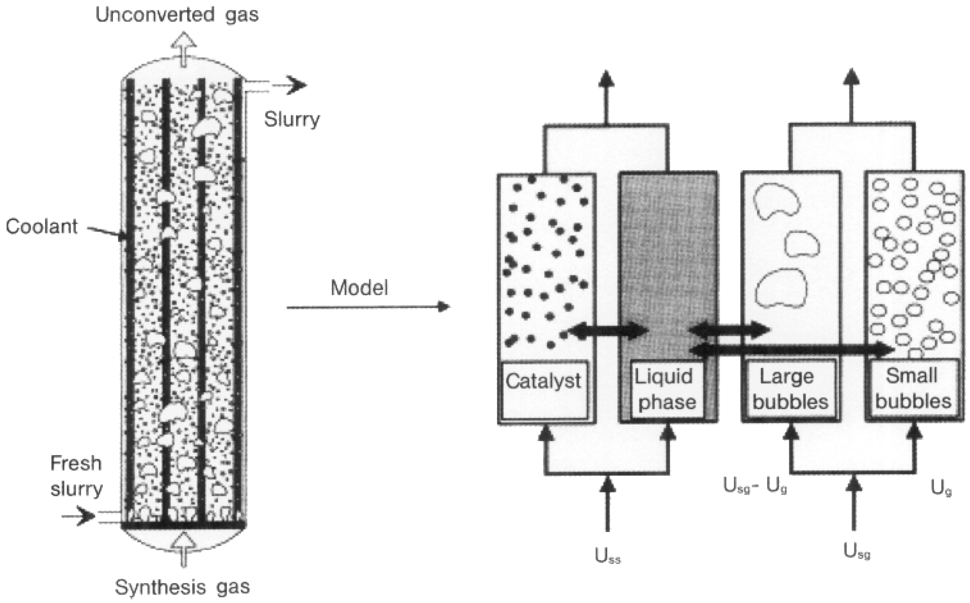


Figure 2.17 Hydrodynamic model of slurry bubble column reactor (SBCR) in the heterogeneous flow regime (Van der Laan et al. 1999)

A highly exothermic reaction takes place on the Fe-based catalytic surface at high temperature (about 250 °C):

$$\begin{aligned} &\mathrm{CO} + \left(1 + \frac{m}{2n}\right)\mathrm{H_2} \rightarrow \frac{1}{n}\mathrm{C}_n\mathrm{H}_m + \mathrm{H_2O} + 165\,\mathrm{kJ/mol} \\ &\mathrm{CO} + \mathrm{H_2O} \leftrightarrow \mathrm{CO_2} + \mathrm{H_2} + 41\,\mathrm{kJ/mol} \end{aligned} \tag{135}$$

where n is the average length of the hydrocarbon chain and m is the number of hydrogen atoms per carbon. Since hydrogen is considered the limiting component, balance equations can be set up for hydrogen only (De Swart et al. 2002). The complex hydrodynamics of gas bubbles are simplified by gas holdups ($\varepsilon_{\mathrm{g,big}}$) of large bubbles ($d_{g,big} = 20\text{–}80\ mm$) and those ($\varepsilon_{\mathrm{g,small}}$) of small bubbles ($d_{g,small} = 1\text{–}6\ mm$), as shown in Fig. 2.18. The dimensionless mass and energy balances for hydrogen are described for the three phases:

- gas phase

$$\begin{aligned} \varepsilon_{\mathrm{g,big}}\frac{\partial \gamma_{\mathrm{big}}}{\partial \tau} = &-\frac{1+\alpha_{\mathrm{cont}}}{(1+\alpha_{\mathrm{cont}}\cdot\gamma_{\mathrm{big}})^2}\frac{(u_{\mathrm{sg}}-u_{\mathrm{df}})}{u_{\mathrm{g0}}}\frac{\partial \gamma_{\mathrm{big}}}{\partial \xi} \\ &+ \frac{\varepsilon_{\mathrm{g,big}}}{Pe_{\mathrm{g,big}}}\frac{\partial^2 \gamma_{\mathrm{big}}}{\partial \xi^2} - St_{\mathrm{g,big}}\left(\gamma_{\mathrm{big}} - x\right) \end{aligned} \tag{136}$$

$$\varepsilon_{\mathrm{g,small}}\frac{\partial \gamma_{\mathrm{small}}}{\partial \tau} = -\frac{u_{df}}{u_{\mathrm{g0}}}\frac{\partial \gamma_{\mathrm{small}}}{\partial \xi} + \frac{\varepsilon_{\mathrm{g,small}}}{\mathrm{g,small}}\frac{\partial^2 \gamma_{\mathrm{big}}}{\partial \xi^2} - St_{\mathrm{g,small}}\left(\gamma_{\mathrm{small}} - x\right) \tag{137}$$

- liquid phase

$$\begin{aligned} \varepsilon_L\frac{\partial x}{\partial \tau} = &-\frac{u_{\mathrm{ss}}}{u_{\mathrm{g0}}}\frac{\partial x}{\partial \xi} + \frac{\varepsilon_L}{Pe_{\mathrm{L}}}\frac{\partial^2 x}{\partial \xi^2} + St_{L,\mathrm{big}}\left(\gamma_{\mathrm{big}} - x\right) \\ &+ St_{L,\mathrm{small}}\left(\gamma_{\mathrm{small}} - x\right) - C_s Da\cdot x\,\mathrm{e}^{-(\gamma/\theta)} \end{aligned} \tag{138}$$

- solid phase

$$C_s(\xi) = \overline{C}_s\frac{Bo\cdot \mathrm{e}^{-Bo\xi}}{1-\mathrm{e}^{-Bo}} \approx 0.25 \tag{139}$$

The dimensionless energy balance for the slurry phase is:

$$\varepsilon_L\frac{\partial \theta}{\partial \tau} = -\frac{u_{\mathrm{ss}}}{u_{\mathrm{g0}}}\frac{\partial \theta}{\partial \xi} + \frac{\varepsilon_L}{Pe_H}\frac{\partial^2 \theta}{\partial \xi^2} - St_H(\theta - 1) + C_s BeDa\cdot x\,\mathrm{e}^{(\gamma/\theta)} \tag{140}$$

The dimensionless variables are denoted $\gamma_{big} = C_{H_2,g,big}/C_{H_2g0}$, $\gamma_{small} = C_{H_2,g,small}/C_{H_2,g0}$, $x = mC_{H_2,L}/C_{H_2,g0}$, $\theta = T/T_c$, $\xi = h/H$ and $\tau = tu_{g0}/H$, where the initial hydrogen concentration $C_{H_2,g0} = 0.38412\ kmol/m^3$, the distribution coefficient of hydrogen between gas and liquid phases $m = 5.095$, the heat-exchanger wall temperature $T_c = 501\ K$, the slurry reactor height $H = 30\ m$, and the inlet gas velocity $u_{g0} = 0.14\ m/s$ are preliminarily given for simulation.

For the gas phase mass balance, the gas holdup of large bubbles ($\varepsilon_{\mathrm{g,big}}$) can be estimated by the following relation with the gas superficial velocity (u_{sg} = 0.14 *m/s*), the small bubble superficial rising velocity (u_{df}), the gas density ($\varrho_{\mathrm{g}} \sim 7$ *kg/m*^{3} at $P \sim 40$ *atm* and $T \sim 500$ *K*), and the reference gas density ($\varrho_{\mathrm{g}}^{ref} \sim 1.3$ *kg/m*^{3} at $P = 1$ *atm* and $T = 293$ *K*):

$$\varepsilon_{\mathrm{g,big}} = 0.3(u_{sg} - u_{df})^{0.58}\left(\frac{\rho_{\mathrm{g}}}{\rho_{\mathrm{g}}^{\mathrm{ref}}}\right)^{0.5} \tag{141}$$

The gas holdup of the small bubbles is given where the transition from the homogeneous to the churn turbulent flow regime occurs (van der Lann et al. 1999):

$$\varepsilon_{\mathrm{g,small}} = \varepsilon_{df}^{\mathrm{ref}}\left(\frac{\rho_{\mathrm{g}}}{\rho_{\mathrm{g}}^{\mathrm{ref}}}\right)^{0.48}\left(1 - \frac{0.7\varepsilon_p}{\varepsilon_{\mathrm{df}}^{\mathrm{def}}}\right) \tag{142}$$

where the small bubble holdup in solids-free liquid is $\varepsilon_{\mathrm{ef}}^{ref} = 0.27$ and the solid holdup is given as $\varepsilon_{\mathrm{p}} = 0.25$. The gas contraction factor is assumed to be $\alpha_{\mathrm{cont}} \sim -0.5$ in Eq. (136). The gas phase Peclet numbers for large and small bubbles are assigned to be $Pe_{g,big} \sim 100$ and $Pe_{g,small} = u_{\mathrm{g0}}G/E_{\mathrm{L}} \sim 80$, respectively. The Stanton numbers of the large bubbles ($St_{g,big} = k_{L,H_2,big}\alpha_{\mathrm{big}}H/m/u_{\mathrm{g0}} \sim 4.51$) and the small bubbles ($St_{g,small} = k_{L,H_2,small}\alpha_{\mathrm{small}}H/m/u_{\mathrm{g0}} \sim 24.7$) are calculated as an empirical correlation proposed by Calderbank and Moo-Young (1961). The superficial velocity of small bubbles, u_{df}, is defined as:

$$u_{df} = u_{\mathrm{g,small}}\varepsilon_{\mathrm{g,small}} \tag{143}$$

where $u_{\mathrm{g,small}} = 2.25\dfrac{\sigma}{\eta_L}\left(\dfrac{\sigma^3\rho_L}{g\eta_L^4}\right)^{-0.273}\left(\dfrac{\rho_L}{\rho_{\mathrm{g}}}\right)^{0.03}$ with liquid viscosity $\eta_{\mathrm{L}} = 6 \times 10^{-4}$ *Pa · s*, surface tension $\sigma = 0.019$ *Pa · m*, liquid density $\varrho_{\mathrm{L}} = 680$ *kg/m*^{3}, gas density $\varrho_{\mathrm{g}} = 7.0$ *kg/m*^{3} and gravity $g = 9.81$ *m/s*^{2}.

For the liquid phase mass balance, the liquid hold up is determined by:

$$\varepsilon_L = 1 - \varepsilon_p - \left(\varepsilon_{\mathrm{g,big}} + \varepsilon_{\mathrm{g,small}}(1 - \varepsilon_{\mathrm{g,big}})\right) \tag{144}$$

The liquid phase Peclet number ($PE_L = u_{\mathrm{g0}}H/E_L$) is assumed to have the same value as the small bubble Peclet number. The liquid Stanton number for large bubbles and small bubbles are defined as $St_{g,big} = {}_{L,H_2,big}a_{big}H/u_{\mathrm{g0}}$ ($\sim$ 22.97) $St_{g,small} = k_{L,H_2,small}a_{\mathrm{small}}H/u_{\mathrm{g0}}$ ($\sim$ 125.87), respectively. The superficial slurry velocity is equal to $u_{ss} = 0.01$ *m/s* in the simulation and the average catalyst concentration fraction is $\overline{C}_s = 0.25$. The Damkohler number as the dimensionless pre-exponential kinetic factor is defined as:

$$Da = Am\varepsilon_L H/u_{\mathrm{g0}} \tag{145}$$

where the preexponential kinetic factor (or collision frequency factor) is $A = 5.202 \times 10^{10}$ *s*$^{-1}$. The Arrhenius number is given from the kinetic data (De Swart et al. 2002):

$$\gamma = \frac{E_a}{RT_c} = \frac{1.175 \times 10^5\,\text{J/mol}}{8.314\,\text{J/mol/K} \cdot 501\,\text{K}} = 28.209 \qquad (146)$$

For the liquid phase energy balance, the heat transfer Peclet number $(Pe_H = \frac{u_{g0}H}{E_L} \approx 7Q)$the heat transfer Stanton number $(St_H = \frac{\alpha_{\text{eff}}a_w H}{\rho_s C_{ps} u_{g0}} \approx 7.0)$, and the dimensionless heat reaction $(Be = \frac{-\Delta H_R C_{H_2,g0}}{m\rho_s C_{ps} T_c} \approx 0.0488)$ are from De Swart et al. (2002).

The model described by the above set (i.e., Eqs. (136)–(140)) of partial differential equations (PDEs), which include convection, diffusion and reaction, is solved with initial conditions and Danckwert's boundary conditions (see Eq. (3)). De Swart et al. (2002) solved the above model numerically using the MOL with finite difference method (FDM) and a BDF ODE integrator. We here use the noniterative CE/SE method (see Section 2.4.3) to solve the model. Figure 2.18 shows dynamic contours

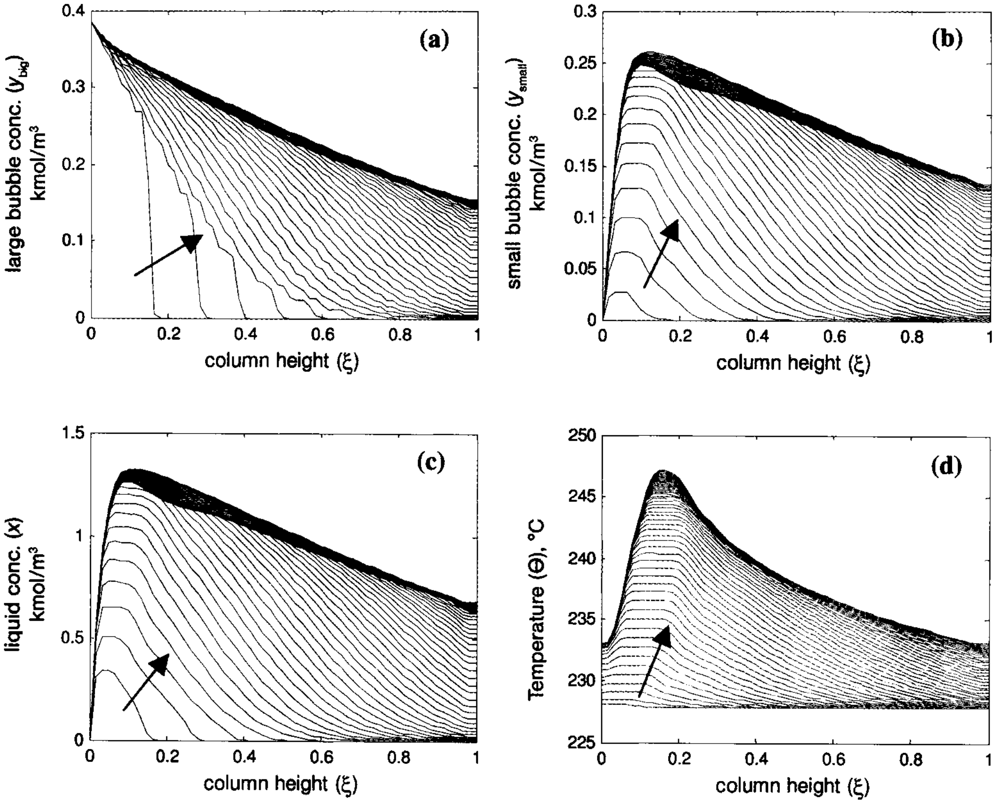


Figure 2.18 Unsteady-state concentration contour of (a) large bubble gas concentration; (b) small bubble gas concentration; (c) liquid concentration; and (d) slurry temperature with respect to the reactor height, within 5 min

of large bubbles, small bubbles and liquid concentrations of hydrogen, and bed temperatures along the reactor height. These profiles show how to reach steady state within 5 min.

From the model-based dynamic simulation we can predict conversion ratio and temperature changes with feed composition, heat-exchanger temperature, feed flow rate, and catalyst types. The optimal operating conditions can be obtained for a given objective function (e.g., cost-benefit function) using the method of nonlinear programming (NLP, see also Section 2.4).

The three-phase bubble column shows complex hydrodynamics of reactant gas bubbles at elevated pressures (e.g., 10–40 atm). Several recent publications have established the potential of computational fluid dynamics (CFD) for describing the hydrodynamics of bubble columns (Krishna and van Baten 2001). Using a commercial CFD code (CFX, AEA Tech., UK) to solve mass/momentum conservation equations in the three phases, Krishna and van Baten (2001) predict the gas holdup and the liquid velocity within a cylindrical two-dimensional reactor at different column dimensions, pressures, and superficial gas velocity. The empirical correlations of the gas holdups in Eqs. (141) and (142) can be verified or predicted for different column dimensions by the CFD simulation results (Krishna et al. 2000). In Section 2.7, we will present in detail a combination of process simulation and CFD.

2.6.4 Population Balance Equation

The population balance equation (PBE) has been demonstrated to describe the particle size distribution (PSD) in various chemical/biological engineering problems such as crystallization, polymerization, emulsion, and microbial cultures. Indeed, modeling with the PBE provides a good description for parameter identification, which may be used for determination of operating conditions, and for process design and control. In crystallization processes, the PBE, which governs the crystal size distribution (CSD), is solved together with mass/energy balances and crystallization kinetics such as nucleation, crystal growth, breakage, and agglomeration. The system, which often leads to hyperbolic-like integro-partial differential equations (IPDEs), is complex due to a lot of feedback relationships between the equations (Wey 1985). To determine the CSD, all equations (e.g., PBE, mass, and energy balances) must be solved simultaneously. An inaccurate solution of a PBE will affect particle nucleation and subsequently particle growth and results in an incorrect CSD. Therefore, a numerical procedure to obtain an accurate solution of PBEs is necessary (Lim et al. 2002).

The crystal size distribution (CSD) is usually expressed as the crystal number (N, no.) or number density (n, no./m, or no./m^3) with respect to the crystal size (L, m) or volume (v, m^3). A simple relationship between the crystal number (N) and the crystal number density (n) is given as follows using the finite volume approach:

$$N_i = \int_{L_i}^{L_{i+1}} n\,\mathrm{d}L \approx n_i(L_{i+1} - L_i) \quad \text{or} \quad N_i = \int_{v_i}^{v_{i+1}} n\,\mathrm{d}v \approx n_i(v_{i+1} - v_i) \tag{147}$$

Both bases (i.e., N and n) can give a good description of the CSD. However, the CSD based on the number (N) is often preferred, for conservation of the mass and the number of crystals, in the cases involving agglomeration and breakage kinetics (Kumar and Ramkrishna 1997).

A number-based PBE as a governing equation of the CSD is usually described in terms of the birth of nuclei, their growth, agglomeration, and breakage:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{nucleation}} + \left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{growth}} + \left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{agglomeration}} + \left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{breakage}} \tag{148}$$

where

$$\left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{nucleation}} = \begin{cases} B_N, & i = 1, \\ 0, & i \neq 1 \end{cases} \tag{149}$$

$$\left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{growth}} = -\frac{\mathrm{d}(G_i \cdot N_i)}{\mathrm{d}L}, \quad i = 1 \ldots N_{\text{mesh}} \tag{150}$$

$$\left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{agglomeration}} = \frac{1}{2}\sum_{j=1}^{i} \beta(L_j, L_{i-j+1}) \cdot N_j \cdot N_{i-j+1} \tag{151}$$

$$\left(\frac{\mathrm{d}N_i}{\mathrm{d}t}\right)_{\text{breakage}} = -\, N_i \sum_{k=1}^{N_{\text{mesh}}} \beta(L_i, L_k) \cdot N_k, \quad i = 1 \ldots N_{\text{mesh}}$$

$$\begin{cases} \gamma_0 \Delta L \cdot \displaystyle\sum_{j=i+1}^{N_{\text{mesh}}} \frac{L_j^a}{L_j} \cdot N_j, & i = 1 \\ \gamma_0 \Delta L \cdot \dfrac{L_i^a}{L_i} \cdot N_i + 2\gamma_0 \Delta L \cdot \displaystyle\sum_{j=i+1}^{N_{\text{mesh}}} \frac{L_j^a}{L_j} \cdot N_j - \gamma_0 L_i^a \cdot N_i, & i = 2 \ldots (N_{\text{mesh}} - 1) \\ \gamma_0 \Delta L \cdot \dfrac{L_i^a}{L_i} \cdot N_i - \gamma_0 L_i^a \cdot N_i, & i = N_{\text{mesh}} \end{cases} \tag{152}$$

Here, B_N is the nucleation function at a minimum nuclei size L_1, G_i is the growth rate, β is called the agglomeration efficiency or the agglomeration kernel, and γ_0 and a are breakage function parameters to be determined. Applying a uniform binary breakage function and a power-law breakage kernel depending on the size (i.e., $\gamma(L_i) = \gamma_0 L_i^a$), Eq. (152) is obtained (Hill and Ng 1995). Equation (148) is a scalar hyperbolic PBE with source terms and it involves only a convection term $\left(\frac{\partial(GN)}{\partial L}\right)$ without any other spatial derivatives. If the PBE of Eq. (148) has a unique growth rate (G), the method of characteristics (MOC) can be applied since the mesh velocities (L_i) are equal to the growth rates (G_i).

2.6.4.1 Method of Characteristics

It is well known that for the scalar linear conservation law (e.g., PBE considered here) there usually exists a unique characteristic curve along which information propagates. If the solution moves along the path line of propagation, the convection term $\frac{\partial(GN)}{\partial L}$ in the PBE disappears. Hence, numerical error and instability caused by approximation of the convection term is removed.

Kumar and Ramkrishna (1997) derived a modified MOC formulation for the PBE:

$$\frac{dN_i}{dt} \equiv 0 = \left(\frac{dN_i}{dt}\right)_{\text{nucleation}} + \left(\frac{dN_i}{dt}\right)_{\text{agglomeration}} + \left(\frac{dN_i}{dt}\right)_{\text{breakage}} \tag{153a}$$

$$\frac{dL_i}{dt} = G_i \tag{153b}$$

where Eq. (153b) is the mesh movement equation. The MOC formulation is numerically solved by using the MOL. To overcome the nucleation problem, a new mesh of the nuclei size (L_1) is added at given time levels. The system size can be kept constant by deleting the last mesh at the same time levels. Since the number of crystal nuclei can vary with the number of mesh points added or deleted, a proper number of added mesh points should be selected according to stiffness of nucleation.

Suppose that a stiff nucleation takes place only at a minimum crystal size ($L_1 = 10^{-5}$) as a function of time:

$$n(t, L_1) = 100 + 10^6 \exp\left(10^{-4}(t - 0.215)^2\right) \tag{154}$$

Within the size range $10^{-5} \leq L \leq 2.0$, the nuclei grow and the crystals aggregate as well as break for $0.0 \leq t \leq 0.5$. A square initial condition as seeds is also given:

$$\begin{aligned} &n(0, L) = 100, \text{ for } 0.4 \leq L \leq 0.6 \\ &n(0, L) = 0.01, \text{ elsewhere} \end{aligned} \tag{155}$$

The kinetic parameters are given: $G = 1$ (linear growth rate), $\beta = 1.5 \times 10^{-5}$ (constant agglomeration kernel) and $\gamma = 1.0 \times L^2$ (breakage kernel). See Lim et al. (2002) for details. The discretized PBE based on the crystal number (N_i) or the crystal density (n_i) is solved by using the implicit BDF ODE integrator in the framework of the MOL.

2.6.4.2 Nucleation and Growth

When the PBE with the nucleation and growth terms is considered on the basis of the crystal density (n), its analytic solution is derived from the MOC:

$$n(t, L) = 100 + 10^6 \exp\left(-10^4((Gt - L) - 0.215)^2\right), \text{ for } 0.0 \leq L \leq Gt, \tag{156a}$$

$$n(t, L) = 100, \text{ for } 0.4 \leq (L - Gt) \leq 0.6, \tag{156b}$$

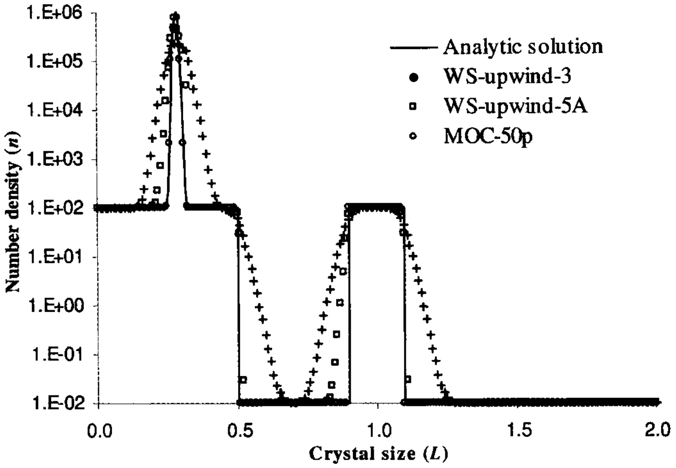


Figure 2.19 CSDs for the stiff nucleation case without agglomeration and breakage

$$n(t, L) = 0.01, \text{ elsewhere} \tag{156c}$$

In the solution, a discontinuous front (due to square seed) and a narrow wave (originating from nucleation) move along the propagation path line, $L = L_1 + Gt$. The numerical tests are carried out on the 200 fixed grids for both the numerical MOC and the WENO schemes. In Fig. 2.19, the numerical results of the WS-upwind-3/5A (see Lim et al. (2002) for details) and MOC-50p (i.e., numerical MOC with an additional 50 mesh points) are compared to the analytic solution Eq. (155) at the end time ($t = 0.5$). While moving fronts are smeared near discontinuities using the WENO schemes on the weighted stencil, the numerical MOC-50p shows a quite good resolution even at the discontinuous fronts.

2.6.4.3
Nucleation, Growth, Agglomeration, and Breakage

When the agglomeration and breakage kinetics are added to the previous PBE, the analytic solution can not be derived. The numerical solution of Eqs. (148) or (153) is obtained on 101 points of the uniform grid, using MOC-20p or WS-upwind-5. Employing the MOC-20p (inserting/deleting 20 mesh points), the following mesh equations are used:

$$\frac{dL_1}{dt} = 0, \quad \frac{dL_i}{dt} = 1 \quad \text{for } i = 2 \ldots 101 \tag{157}$$

In Fig. 2.20, CSD changes obtained from MOC-20p are depicted according to the kinetics used. The solid line is the analytic solution for the pure growth problem without agglomeration and breakage. Since numerical diffusion error is small, high resolution is observed at the corners of steep fronts. Due to the agglomeration term, the CSD spreads out and the population of large crystal sizes increases (see Fig.

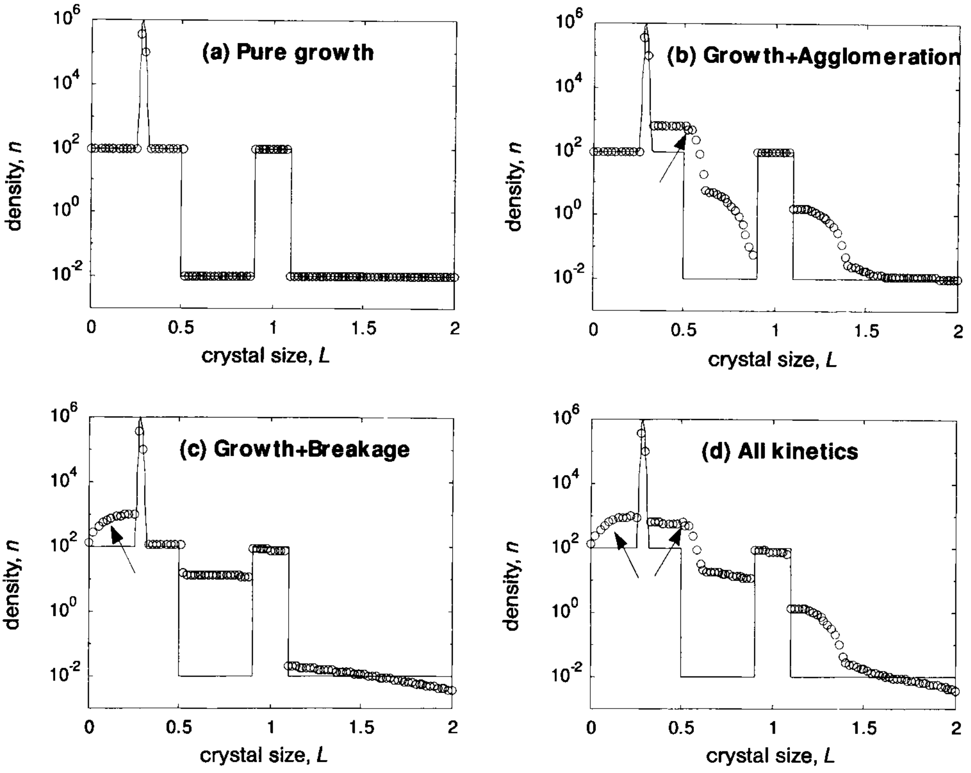


Figure 2.20 CSD changes obtained by the MOC-20p on 102 meshes according to the growth, agglomeration, and breakage terms (*solid line*: analytic solution for pure growth)

2.20b). In contrast, the breakage term increases the population of small crystal sizes (see Fig. 2.20c). The CSD of the PBE with four kinetics is dispersed more broadly, as shown in Fig. 2.20d.

Using the fifth-order WENO scheme (WS-upwind-5), Fig. 2.21 also shows effects of the growth, agglomeration, and breakage terms on the CSD. Considerable numerical dissipation is found in steep regions (or discontinuities) in Fig. 2.21, and also shown in Fig. 2.19. However, comparing Fig. 2.20d with Fig. 2.21d, the two solutions are similar due to the effects of agglomeration and breakage on the CSD.

Though the modified MOC gives more accurate numerical results than the WENO scheme, there are some limitations to using it such as the need for careful determination of adding/deleting time levels and a unique mesh velocity equation (or growth rate, see Eq. (153b)). Using the spatial discretization methods (e.g., MOL with WENO schemes) to circumvent these limitations, attention must be paid to discretization of the growth term (convection term), which can cause much numerical error and instability in the presence of steep fronts or discontinuities.

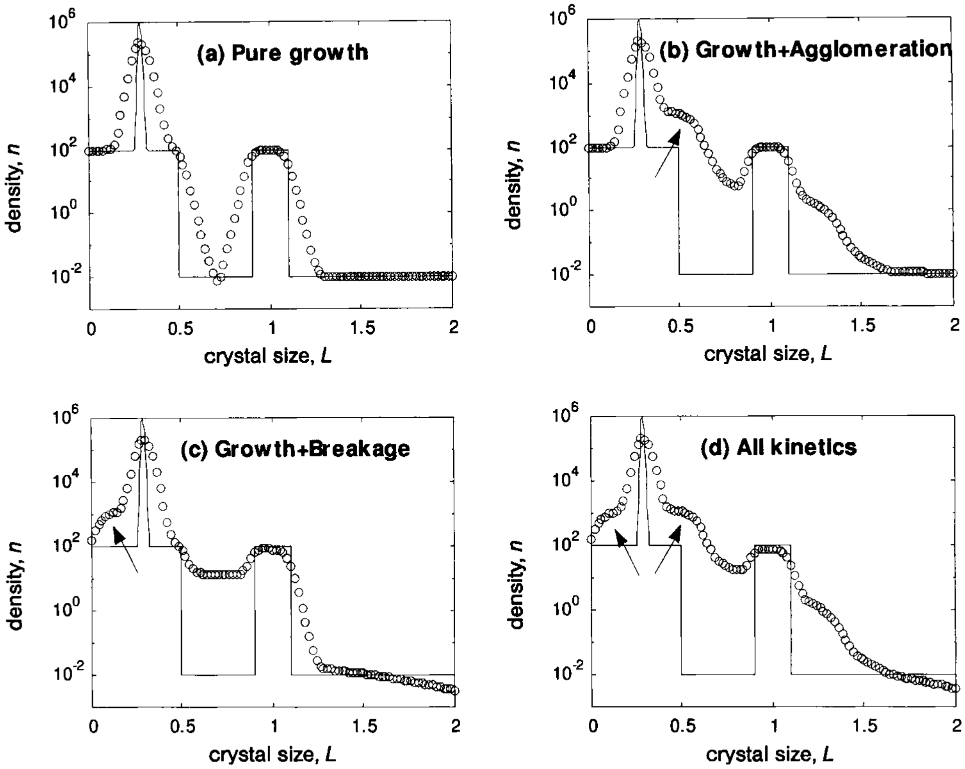


Figure 2.21 CSD changes obtained by the WS-upwind-5 scheme on 101 meshes according to the growth, agglomeration, and breakage terms (*solid line*: analytic solution for pure growth)

2.6.5 Cell Population Dynamics

Cell cultures are composed of discrete microorganisms whose population dynamics play an important role in bioreactor design and control. The cell cultures are known to exhibit autonomous oscillations that affect bioreactor stability and productivity. To increase the productivity and stability, it is therefore desirable to derive a dynamic model that describes the oscillatory behavior and to develop a control strategy that allows modification of such intrinsic reactor dynamics (Henson 2003).

As a model example for cell culture dynamics, consider a segregated/unstructured modeling based on the cell population balance equation (PBE) coupled to metabolic reactions that are relevant for the extracellular environment (Zhu et al. 2000; Mhaskar et al. 2002). The segregated/unstructured model provides a realistic description of the cell cycle events that lead to sustained oscillation in cell cultures under the assumption that oscillations arise as a result of interactions between the cell population and the extracellular environment.

The cell population dynamics including cell growth and cell division is described by a partial differential equation including a convection term $\left(\frac{\partial v_g W}{\partial m}\right)$, newborn-cell birth term ($\int 2\, p\Gamma W dm$), mother-cell division death term ($-\Gamma W$) and dilution loss ($-DW$):

$$\frac{\partial W(m,t)}{\partial t} = -\frac{\partial\left(v_g(S')\cdot W(m,t)\right)}{\partial m} + \int_0^{m'} 2p(m,m')\Gamma(m',S')W(m',t)\,\mathrm{d}m' - \left[D+\Gamma(m)\right]W(m,t) \tag{158}$$

where $W(m, t)$ is the cell number concentration as a function of mass (m) and time (t), $v_g(S'$ is the overall single cell growth rate at the substrate concentration (S'), p(m, m') is the newborn-cell mass distribution function with newborn-cell mass m and mother-cell mass m', $\Gamma(m', S')$, is the division intensity function, and D is the dilution rate. Detail models and their parameters are given below on the basis of Mhaskar et al. (2002) for a *S. cerevisiae* (or yeast) culture. For convenience, all of the masses have the units [$\times\ 10^{-13}$ g] and the cell number concentration ($W\ (m,\ t)$ has the units [$\times\ 10^{-13}$ *no./g*].

The division intensity function is introduced to account for the probability nature of cell division:

$$\Gamma(m,S') = \begin{cases} 0, & m \le m_t^* + m_0 \\ \gamma\cdot \mathrm{e}^{-\varepsilon(m-m_d^*)^2}, & m_t^* + m_0 < m < m_d^* \\ \gamma, & m \ge m_d^* \end{cases} \tag{159}$$

where m^*_t is the cell transition mass, $m_0 = 1$ is the minimum cell mass for division, and m^*_d is the division mass. $\varepsilon = 5$ and $\gamma = 200$ are the constant parameters that determine the transition rate and the maximum intensity value, respectively.

Sustained oscillations are generated through the introduction of a synchronization mechanism in which the transition and division masses are functions of the nutrient concentration. The following saturation functions are used:

$$m_t^*(S') = \begin{cases} m_{t0} + K_t(S_l - S_h), & S' < S_l \\ m_{t0} + K_t(S' - S_h), & S_h \le S' \le S_l \\ m_{t0} & S' > S_h \end{cases} \tag{160a}$$

$$m_d^*(S') = \begin{cases} m_{d0} + K_d(S_l - S_h), & S' < S_l \\ m_{d0} + K_d(S' - S_h), & S_h \le S' \le S_l \\ m_{d0} & S' > S_h \end{cases} \tag{160b}$$

where the substrate concentration is $S' = G' + E'$, and the constants are given as $S_l =$ 0.1 g/l, $S_{h\,=\,2.0\ g/l,\ mt0} = 4.55$, $m_{d0} = 10.75$, $K_t = 0.01\ \cdot\ l^{-1}$ and $K_d = 3.83\ \cdot\ l^{-1}$.

The newborn cell probability function $p(m, m')$ has the form:

$$p(m, m') = \begin{cases} \alpha \cdot \mathrm{e}^{-\beta(m-m_t^*)^2} + \alpha \cdot \mathrm{e}^{-\beta(m-m'+m_t^*)^2}, & m' > m \text{ and } m' > m_t^* + m_0 \\ 0, & \text{elsewhere} \end{cases} \tag{161}$$

Here the constants are set to $\alpha = \sqrt{10/\pi}$ and $\beta = 40$. This function yields two Gaussian peaks in the cell number distribution, one centered at m^*_t corresponding to mother cells and one centered at $m^*_t - m'$ corresponding to daughter cells.

Oscillatory yeast dynamics are observed in glucose-limited growth environments. Under such conditions, both glucose and the excreted product ethanol can serve as substrates for cell growth. The following reaction sequence accounts for the relevant metabolic pathways, glucose fermentation, glucose oxidation, and ethanol oxidation:

$$\begin{aligned} &C_6H_{12}O_6 - 2C_2H_5OH + 2CO_2 \\ &C_6H_{12}O_6 + 6O_2 \rightarrow 6CO_2 + 6H_2O \\ &C_2H_5OH + 3O_2 \rightarrow 3H_2O \end{aligned} \tag{162}$$

The three kinetic rates proposed are:

$$\begin{aligned} K_{\mathrm{gf}}(G') &= \frac{\mu_{\mathrm{mgf}} G'}{K_{\mathrm{mgf}} + G'} \\ K_{\mathrm{go}}(G', O) &= \frac{\mu_{\mathrm{mgo}} G'}{K_{\mathrm{mgo}} + G'} \frac{O}{K_{\mathrm{mgd}} + O} \\ K_{\mathrm{eo}}(G', E', O) &= \frac{\mu_{\mathrm{meo}} E'}{K_{\mathrm{meo}} + E'} \frac{O}{K_{\mathrm{med}} + O} \frac{K_{\mathrm{inhibit}}}{K_{\mathrm{inhibit}} + G'} \end{aligned} \tag{163}$$

where G' and E' represent intracellular glucose and ethanol concentrations, respectively, and O is the dissolved oxygen concentration. $\mu_{\mathrm{mgf}} = 30(\times 10^{-13}\ g/h)$, $\mu_{\mathrm{mgo}} = 3.25\ (\times\ 10–13\ g/h)$, and $\mu_{\mathrm{meo}} = 7(\times\ 10^{-13}\ g/h)$ are maximum consumption rates, $K_{mgf} = 40\ g/l$, $K_{Mgo} = 2\ g/l$, $K_{mgd} = 0.001\ g/l$, $K_{meo} = 1.3\ g/l$, and $K_{med} = 0.001\ g/l$ are saturation constants, $K_{inhibit} = 0.4\ g/l$ is a constant that characterizes the inhibitory effect of glucose on ethanol oxidation. The overall single cell growth rate $v_g(S')$ is the sum of the growth rates due to the three metabolic reactions.

$$v_{\mathrm{g}}(S') = K_{\mathrm{gf}}(G') + K_{\mathrm{go}}(G', O) + K_{\mathrm{eo}}(G', E', O) \tag{164}$$

For intracellular glucose and ethanol concentrations (G', E') and liquid oxygen concentrations (O) in Eq. (163), the mass balance equations of these substrates are

$$\frac{\mathrm{d}G'}{\mathrm{d}t} = k_{\mathrm{g}}(G - G') \tag{165}$$

$$\frac{\mathrm{d}E'}{\mathrm{d}t} = k_{\mathrm{e}}(E - E') \tag{166}$$

$$\frac{\mathrm{d}O}{\mathrm{d}t} = k_{\mathrm{o}}a(O^* - O) - \left(\frac{192}{180}\frac{K_{\mathrm{go}}(G')}{Y_{\mathrm{go}}} + \frac{96}{46}\frac{K_{\mathrm{eo}}(E')}{Y_{\mathrm{eo}}}\right) N_{\mathrm{total}} \tag{167}$$

where G and E are extracellular concentrations, k_g and $k_e = 20\ h{-}1$ are glucose and ethanol uptake rates, respectively, $k_oa = 1500\ h^{-1}$ is the oxygen mass transfer rate, and $Y_{go} = 0.65\ g/g$ and $Y_{eo} = 0.5\ g/g$ are the yield coefficients in the glucose oxidation and ethanol oxidation reactions, respectively. The total cell number (N_{total}) of microorganisms is defined by

$$N_{\text{total}} = \int_0^\infty W(m,t)\,\mathrm{d}m \tag{168}$$

The saturation oxygen concentration, O^*, is obtained from the oxygen solubility, which is assumed to be governed by Henry's law: $O^* = H_o \frac{RT}{M_{w,O_2}} O_{out}$ with the Henry's rate constant ($H_o = 0.0404\ g/l/atm$), the gas constant ($R = 0.082057\ l \cdot atm/mol/K$), the absolute temperature ($T = 298\ K$), the molecular weight of O_2 ($M_{w,O_2} = 32$) and the oxygen concentration in the gas exhaust stream (O_{out}). The gas phase oxygen balance is:

$$V_g \frac{\mathrm{d}O_{\text{out}}}{\mathrm{d}t} = F(O_{\text{in}} - O_{\text{out}}) - k_oa(O^* - O) \cdot V_l \tag{169}$$

where $V_g = 0.9\ l$ and $V_l = 0.11$ are the gas phase and liquid phase volumes, respectively, $F = 90\ l/h$ is the volumetric air-feed flow rate, and $O_{in} = 0.275\ g/l$ (= 0.21 atm) is the oxygen concentration in the air-feed stream.

For extracellular glucose and ethanol concentrations (G and E), the substrate mass balance equations are:

$$\frac{\mathrm{d}G}{\mathrm{d}t} = D(G_f - G) - \left(\frac{K_{gf}(G')}{Y_{gf}} + \frac{K_{go}(G')}{Y_{go}}\right) N_{\text{total}} \tag{170}$$

$$\frac{\mathrm{d}E}{\mathrm{d}t} = D(E_f - E) + \frac{92}{180}\frac{K_{gf}(G')}{Y_{gf}} N_{\text{total}}^{\text{ethanol}} - \frac{K_{eo}(E')}{Y_{eo}} N_{\text{total}} \tag{171}$$

where $D = 0.18\ h^{-1}$ is the dilution rate, $G_f = 30\ g/l$ and $E_f = 0\ g/l$ are the feed glucose and ethanol concentrations, respectively, and $Y_{gf} = 0.15\ g/g$ is the yield coefficient in the glucose fermentation reaction. The total cell number for microorganisms related to ethanol excretion is denoted in Eq. (171) as:

$$N_{\text{total}}^{\text{ethanol}} = \int_0^\infty f(m) W(m,t)\,\mathrm{d}m \tag{172}$$

Experimental data suggests that key products, such as ethanol, are excreted primarily by budding cells. This behavior is modeled by $f(m)$:

$$f(m) = \begin{cases} 0, & m \le m_t^* \\ \gamma_e\, e^{-\varepsilon_e (m - m_t^* - m_e)^2}, & m > m_t^* \end{cases} \tag{173}$$

where $\gamma_e = 1.25$, $\varepsilon_e = 15$ and $m_e = 1.54$ are constant parameters.

The liquid phase carbon dioxide balance (C) is

$$\frac{\mathrm{d}C}{\mathrm{d}t} = k_c a(C^* - C) + \left(\frac{264}{180}\frac{K_{go}(G')}{Y_{go}} + \frac{88}{46}\frac{K_{eo}(E')}{Y_{eo}}\right) N_{\text{total}} + \frac{88}{180}\frac{K_{gf}(G')}{Y_{gf}} N_{\text{total}}^{\text{ethanol}} \quad (174)$$

where $k_c a = 1500\ h^{-1}$ is the CO$_2$ mass transfer rate and C^* is the saturation CO$_2$ concentration modeled by $C^* = H_c \dfrac{RT}{M_{w,CO_2}} C_{out}$ with the Henry's rate constant (H_c = 1.48 $g/l/atm$ at pH = 5.0), the molecular weight of CO$_2$ ($M_{w,CO_2} = 44$) and the CO$_2$ concentration in the gas exhaust stream (C_{out}). The gas phase CO$_2$ balance is:

$$V_g \frac{\mathrm{d}C_{\text{out}}}{\mathrm{d}t} = F(C_{\text{in}} - C_{\text{out}}) - k_c a(C^* - C) \cdot V_l \quad (175)$$

where $C_{\text{in}} = 0.00054\ g/l$ (= 0.0003 atm) is the carbon dioxide concentration in the air-feed stream.

In summary, this cell PBE model is described by a PDAE system containing one PDE for the cell population ($W(m, t)$) and eight ODEs for eight substrate variations (G, E, G', E', O, C, O_{out}, and C_{out}). The single cell growth rate ($v_g(G', E', O)$) is computed in Eq. (164).

The initial condition of $W(m, t)$ is set to $W(m, 0) = 0.5\ e{-}5\cdot(m{-}6)^2$, $1 \leq m \leq 11$. The boundary condition of $W(m, t)$ is also given as $W(11, t) = 0$ for $0 \leq t \leq 6\ hr$. For the eight-substrate concentrations, their initial values are $G' = G = 0.8$, $E' = 0.01$, $E = 0.0001$, $O = 0.008$, $C = C_{out} = 0.003$, and $O_{out} = 0.275$.

For the solution of cell PBE models, Zhu et al. (2000) and Mhaskar et al. (2002) used the orthogonal collocation FEM (Finlayson 1980). Motz et al. (2002) reported that the CE/SE method gives better performance in terms of accuracy and computational time than a flux limited finite volume method. Mantzaris et al. (2001a,b) compared several numerical methods such as finite difference methods (FDM) with explicit/implicit time integration and finite element methods (FEM) with explicit/implicit time integration, where they suggested (i) the time-explicit scheme (e.g., Runge-Kutta method) is better for computational-efficiency than the implicit time-integration scheme (e.g., BDF-types) and (ii) the finite difference method is preferred for multidimensional PBEs to the finite element method due to computational efficiency.

Fortunately, this model can be solved by the modified MOC (see Eq. (153)), since there is a unique growth rate Eq. (164). The integration terms appearing in Eqs. (158), (168), and (172) are simply evaluated by the Trapezoidal rule:

$$\int f(m)\,\mathrm{d}m \approx \frac{1}{2}\sum_{i=1}^{N_{\text{mesh}}-1} \left(f(m_{i+1}) + f(m_i)\right) \cdot (m_{i+1} - m_i) \quad (176)$$

where N_{mesh} is the number of mesh points.

When the numerical MOC is used in the framework of the MOL, it is not easy to provide its Jacobian full matrix with hand-coding due to strong nonlinearity and it will be prohibitive to numerically evaluate the Jacobian because of the large system size (i.e., the number of equations is ($2 \times N_{mesh} + 8$), and the Jacobian matrix is 2 ×

$N_{mesh} + 8)^2$). The automatic differentiation technique is appreciated in this case for accuracy and computation efficiency.

Figure 2.22 shows the dynamics of the cell population number density ($W(m, t)$) is solved by the numerical MOC on 82 mesh points (N_{mesh}) where 135 mesh points are added at $m = 1$ and also deleted at $m = 11$. Due to cell division, the cell number density of small sizes tends to increase with time and the oscillatory behavior of the cell number density are regularized after about $t = 6$ *hr*.

The oscillatory behaviors of the cell number (N_{total}) in Eq. (168) and the cell mass $\left(= \sum_{i=1}^{N_{\text{mesh}-1}} N_i \frac{(m_i + m_{i+1})}{2}\right)$ are shown in Figs. 2.23a and b. The cell number variation affects the extracellular glucose/ethanol (G and E) and oxygen/carbon dioxide (O_{out} and C_{out}) concentrations in the gas exhaust stream (see Figs. 2.23c–f, respectively). As ethanol is excreted, primarily by budding cells (see Eq. (173)), it is shown that the extracellular ethanol concentration (E) slowly reaches a regular oscillatory state in Fig. 2.23d. Figures 2.23g,h depict the dynamics of the consumed oxygen concentration ratio $100 \times \left(\frac{O_{in} - O_{out}}{O}\right)$ and evolved carbon dioxide concentration ratio $100 \times \left(\frac{C_{in} - C_{out}}{C}\right)$, respectively.

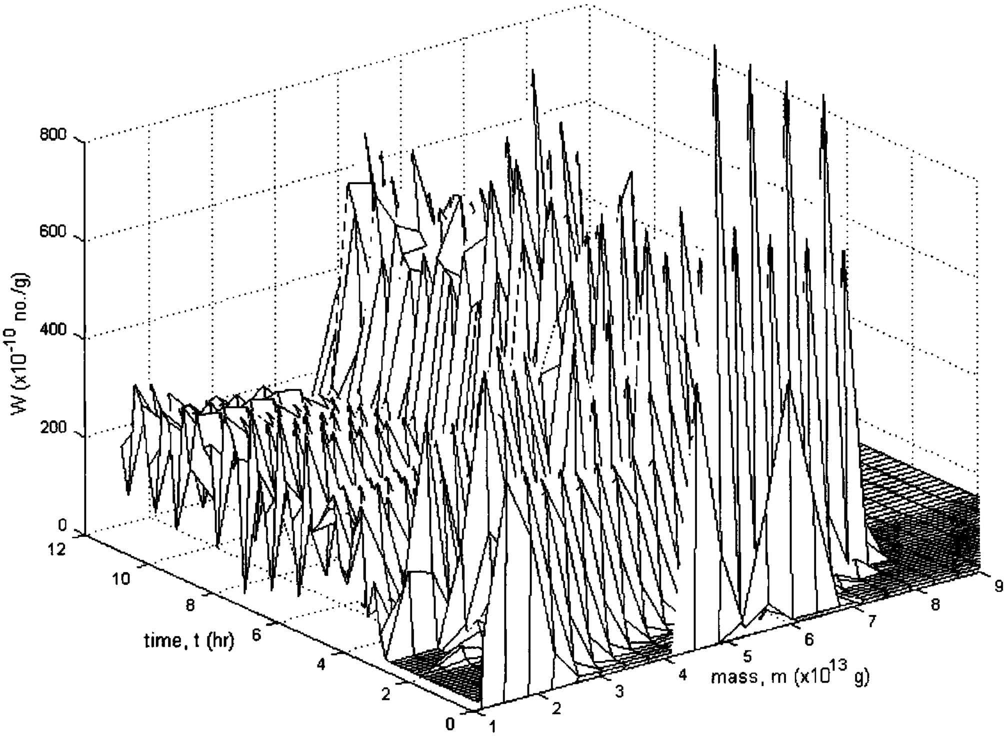


Figure 2.22 Distribution of cell population concentration $W(m,t)$ over mass and time

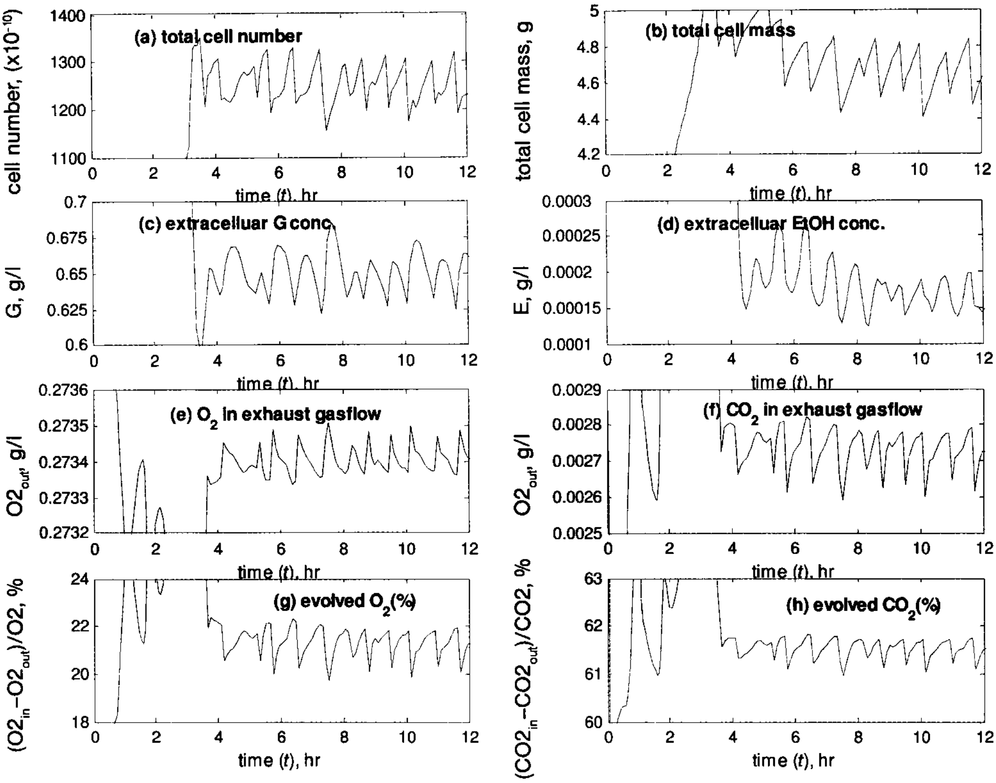


Figure 2.23 Oscillatory behaviors in time of (a) cell number; (b) cell mass; (c) extracellular glucose (G); (d) extracellular ethanol (E); (e) oxygen in exhaust gas stream (O_{out}); (f) carbon dioxide in exhaust gas stream (C_{out}); (g) evolved oxygen ratio; and (h) evolved carbon dioxide ratio

All of the parameters used for this simulation need to be adjusted to experimental data, as shown in Mhaskar et al. (2002).

2.7 Process Model and Computational Fluid Dynamics

A multiscale model is a composite mathematical model formed by combining partial models that describe phenomena at different characteristic length and time scales. For example, modeling of a packed-bed catalytic reactor involves *microscale* chemical kinetics at the active sites on the catalyst, mesoscale transport processes through the pores of the catalyst pellets, and macroscale flow and heat exchange at the reactor vessel level. Computational tools such as molecular dynamics (MD), computational fluid dynamics (CFD), and process simulation have been used to help fill particular length- and timescale gaps. In general, despite their obvious connection, phenomena at different characteristic scales have usually been studied in isolation (see Section 2.5).

One of the key challenges facing process modeling today is the need to describe complex interactions between hydrodynamics and the other physical/chemical phenomena. This is particularly important in the case of complex systems (e.g., polymerization, crystallization, and agitated bioreactors) in which the constitutive phenomena interacts with mixing and fluid flow behavior.

Process simulation tools, which play an increasingly central role within most process engineering activities are able to represent (i) multicomponent, multiphase, and reactive systems, (ii) individual unit operations, multiple interconnected units, or entire plants, and (iii) thermodynamic properties. However, most of the models used by process simulation tools either ignore spatial variations of properties within each unit operation (invoking the well-mixed tank assumption) or are limited to simple idealized geometries. Moreover, the treatment of fluid mechanics is usually quite rudimentary (Bezzo et al. 2003).

CFD techniques solve fundamental mass, momentum, and energy conservation equations (e.g., the Navier-Stokes equation) in complex three-dimensional geometries. From CFD simulation, some valuable information (e.g., mass flow rate, heat transfer coefficient, velocity, etc.) for process simulation can be obtained. However, CFD's ability is still limited in application to complex reactive systems and multiphase processes with multicomponent phase equilibria. Furthermore, performing realistic dynamic simulation often requires excessive computational time. In view of the above, CFD and process simulation technologies are highly complementary (Bezzo et al. 2000). Combination of process simulation and CFD can therefore lead to significant advantages in accurate modeling of processes.

2.7.1
Computational Fluid Dynamics

The CFD technique has focused on the solution of PDEs representing conservation equations describing fluid flow over domains of often complex geometry. There are several commercial CFD packages such as Fluent (Fluent, Inc.), CFX (AEA Tech., Harwell, UK) and FemLab (COMSOL).

The CFD packages usually comprise three distinct elements, namely preprocessing (geometry specification, model selection, parameter specification, and grid generation), numerical solution procedure and post-processing (visualization and data treatment). In the solution procedure, mass/momentum/energy conservation equations are solved within the specified geometry. Generic conservation equations may be described by PDEs with advection, viscosity/diffusivity and source terms:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}\left(\nu\phi - \Gamma_\phi \frac{\partial \phi}{\partial x}\right) - s_\phi = 0 \tag{177}$$

where Φ is a conserved quantity such as mass, energy or momentum, Γ_Φ the viscosity or diffusivity, s_Φ the source or sink and x the set of spatial dimension variables. The models are called the compressible (or incompressible) Navier-Stokes equation. When the viscosity/diffusivity terms can be neglected due to relatively small influence on the result, the inviscid Euler equation is obtained. Let ϱ, u, p, and e be the mass density, velocity, pressure, and energy per unit volume, respectively. The inviscid Euler equation of a perfect gas can be expressed as:

$$u_t - f_x - f_y - f_z = 0 \tag{178}$$

where $u = (\varrho, \varrho u, e)^T$ and $f = (\varrho u, p + \varrho u^2, ue + up)^T$. We may write $e = \varrho e_{int\,ernal} + \frac{1}{2}$ ϱu^2, where $e_{int\,ernal}$ is the internal energy per unit mass. Therefore, this equation is the three-dimensional hyperbolic PDE. Most CFD packages do not use the MOL approach (see Section 2.3) of reducing PDEs into ODEs in time. Instead, they choose to discretize both temporal and the spatial dimensions (i.e., fully discrete methods, see Section 2.4), thereby reducing the PDEs into a set of nonlinear algebraic equations (Oh 1995).

The process simulation models mentioned in Section 2.6 represent specific and simplified conservation equations related to the macroscopic process level. Here, complex fluid dynamics is lumped by parameters or simple empirical equations (e.g., the three phase bubble column model in Section 2.6.3). Microscopic chemical reactions are simplified by kinetic equations as a function of temperature, pressure, and concentrations. If we use a full CFD simulation including complex reactions, thermodynamics, population dynamics and hydrodynamics, it would be practically infeasible because of high computing load and lack of existing tools.

To effectively take into account interactions between hydrodynamics and the other physical/chemical phenomena, a hybrid approach, namely, multizonal/CFD simulation (Bauer and Eigenberger 1999 and 2001; Bezzo et al. 2003), is proposed (see Fig. 2.24). Several zones, assumed to be well-mixed compartments, are described by process models (e.g., AEs, ODEs, PDEs or PBEs) with the exception of the fluid-flow ones, and CFD simulation provides each zone with the mass flow rate at interzonal interfaces and additional fluid dynamical properties such as mass transfer coefficient and turbulent energy dissipation rate (Bezzo et al. 2003).

2.7.2
Combination of CDF and Process Simulation

Figure 2.24 shows a structure of the general multizonal/CFD model. The spatial domain of interest is divided into several zones (z_1–z_5). Each single zone (z) is considered to be well mixed and homogeneous. Two zones can interact with each other via an interface that connects a port (p) of one zone with a port of the other. The flow of material and/or energy across each interface is assumed to be bidirectional. The transient behavior of a zone is described by a set of algebraic equations (AEs), ordi-

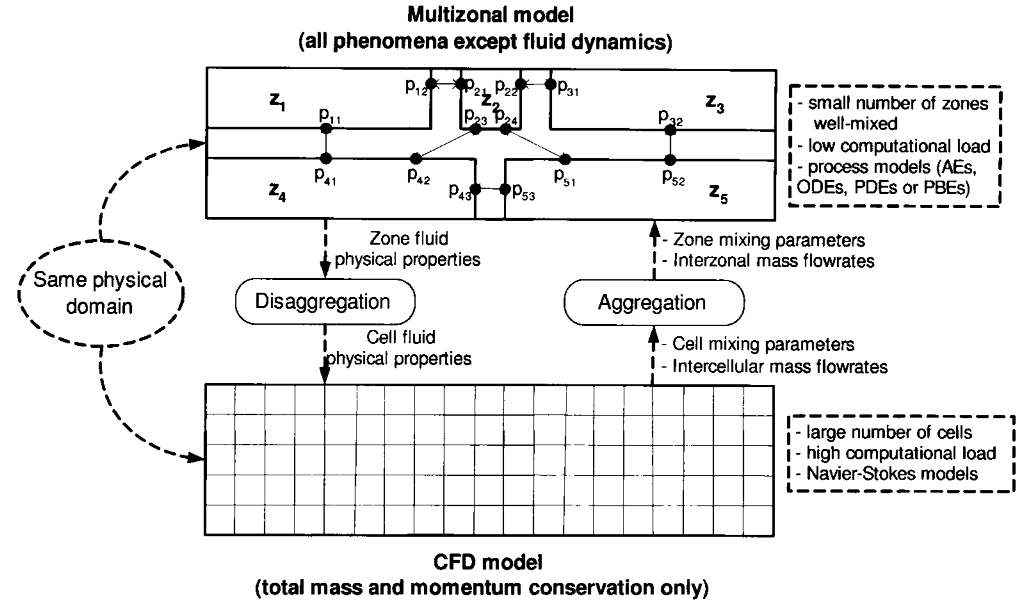


Figure 2.24 Structure of the general multizonal/CFD model (Bezzo et al. 2003)

nary differential equations (ODEs), partial differential equations (PDEs) or population balance equations (PBEs). The multizonal model uses detailed dynamic modeling of all relevant physical phenomena, with the exception of fluid-flow, over a physical domain divided into a relatively small number of zones.

Mixing parameters and interzonal mass flow rates are determined by solving a detailed CFD model over the same physical domain. The CFD model focuses solely on fluid-flow prediction, trying to do this as accurately as possible by dividing the space into a relatively large number of cells and solving the total mass and momentum conservation equations. Thus, the CFD model does not attempt to characterize intensive properties such as composition, temperature or particle size distribution. The transient behavior is ignored, based on the assumption that fluid-flow phenomena operates on a much shorter time scale than all other phenomena.

The solution of the CFD model will require knowledge of the distribution of physical properties (e.g., viscosity, density, compressibility factor, etc.) throughout the physical domain of interest. These properties are usually a function of the system-intensive properties and are computed within the multizonal model.

The hybrid model is formed by the coupling of the multizonal model with the CFD mode, both representing the same spatial domain. The mapping between the zone and cell is achieved by means of appropriate disaggregation and aggregation procedure (Bezzo et al. 2003).

The multizonal concept and CFD simulation is applied to bubble column reactors (Bauer and Eigenberger 1999, 2001) and to a bioreactor processing and mixing a highly viscous fluid (Bezzo et al. 2003).

2.8 Discussion and Conclusion

In many chemical and biotechnical processes, partial derivatives result from a consequence of dynamic behaviors of mass, energy and momentum in space. The derivative or algebraic terms describe fluid flow, physical phenomena and/or constitutive relations in the different phases. The description of convective and diffusive (dispersive) fluxes introduces first and second order spatial derivatives. Mass exchanges between the fluid and stationary phases (e.g., reactions and adsorptions) are described by time-dependent differential equations. Equilibrium relations and physical properties are described with algebraic equations (AEs). Thus process models are in general represented as partial differential equations (PDEs) coupled with algebraic equations (AEs), i.e., PDAEs with pertinent initial and boundary conditions.

In this chapter, a large class of one-dimensional PDAE models is presented in Section 2.6 to introduce the most frequently applied numerical methods for their solution (Sections 2.3–2.5). Finally, the complementary relations between process simulation and computational fluid dynamics most often employed to solve models with fully developed flow field are demonstrated (Section 2.7).

A very important element for a correct discretization is to ensure that the discretized formulation (or numerical approximation) indeed converges to the continuous formulation (or physical model), as the discretization (or spatial and/or temporal stepsize) is refined. However, this issue contains several subtleties and is not dealt with in detail in this chapter.

To improve accuracy and efficiency of numerical solutions, it is desirable to select appropriate numerical methods according to the physical models considered. The method of lines (MOL), which includes time integration and spatial discretization, is adequate to solve stiff problems such as diffusion-dominated models and fast reaction problems (Section 2.3). In the presence of steep moving fronts, the MOL incorporating adaptive and moving mesh methods to capture large spatial variations provides efficient solutions (Section 2.5).

When there is a unique solution propagation path line, the method of characteristics (MOC) can be formulated. The MOC formulation can be solved in the framework of the MOL (Section 2.6.4). Since the solution moves along the propagation path line, numerical error and instability caused by approximation of the convection term is avoided.

In solving convection-dominated models, the conservation element and solution element (CE/SE) method is appreciated for accuracy and efficiency due to a finite volume approach and explicit time integration (Section 2.4.3). The CE/SE method possesses low numerical dissipation error but a fine time stepsize is needed for stiff models.

Most CFD packages use fully discrete methods (Section 2.4) rather than the MOL (Section 2.3) for solving conservation laws for fluid flow within an often complex multidimensional geometry. Combination of process simulation and CFD is useful to describe complex interactions between fluid hydrodynamics and other physical/chemical phenomena (Section 2.7).

Although many efficient methods have emerged for solving PDEs/PDAEs, several challenges remain. The numerical method of PDEs with stiff nonlinear source terms is one of the currently active research areas (Ahmad and Berzins 2001; Hyman et al. 2003), where mesh refinement by an efficient control of space-time errors is needed. Simulation methods for the processes considerably influenced by fluid hydrodynamics should be improved by combining CFD technologies. Hybrid dynamic systems that exhibit coupled continuous and discrete behaviors have also attracted much attention (Mao and Petzold 2002). When a reactant within a phase of a spatially distributed reactor disappears and appears, special considerations are required for the hybrid system to be properly handled by the numerical methods.

References

1 *Ahmad I. Berzins M.* MOL solvers for hyperbolic PDEs with source terms, Mathematics and Computers in Simulation 56 (2001) p. 115–125

2 *Ascher U. M. Petzold L. R.* Computer methods for ODEs and DAEs, SIAM, Philadelphia 1998

3 *Ayasoufi A. Keith T. G.* Application of the conservation element and solution element method in numerical modeling of heat conduction with melting and/or freezing 13(4) (2003) p. 448–471.

4 *Bauer M. Eigenberger G.* A concept for multiscale modeling of bubble columns and loop reactors Chem. Eng. Sci. 54 (1999) p. 5109–5117

5 *Bauer M. Eigenberger G.* Multi-scale modeling of hydrodynamics, mass transfer and reaction in bubble column reactors Chem. Eng. Sci. 56 (2001) p. 1067–1074

6 *Berger M. J. Oliger J.* Adaptive mesh refinement for hyperbolic partial differential equations J. Comp. Phy. 53 (1984) p. 484–512

7 *Berger M. J. Leveque R. J.* Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, SIAM J. Numer. Anal. 35(6) (1998) p. 2298–2316

8 *Bezzo F. Macchietto S. Pandelides C. C.* A general framework for the integration of computational fluid dynamics and process simulation, Computers and Chemical Engineering 24 (2000) p. 653–658

9 *Bezzo F. Macchietto S. Pandelides C. C.* General hybrid multizonal/CFD approach for bioreactor modeling AIChE J. 49(8) (2003) p. 2133–2148

10 *Bischof C. Hovland P.* Using ADIFOR to compute dense and sparse Jacobians, Technical memorandum ANL/MCS-TM-159, Mathematics and computer science division, Argonne National Laboratory, USA 1991

11 *Bischof C. Carle A. Corliss G. Griewank A. Hovland P.* ADIFOR: Generating derivative codes from Fortran programs, Scientific Programming 1(1) (1992) p. 11–29

12 *Bischof C. Carle A. Khademi P. Mauer A. Hovland P.* ADIFOR 2.0 user's guide, Technical report ANL/MCS-TM-192, Mathematics and computer science division, Argonne National Laboratory, USA 1998

13 *Calderbank P. H. Mooyoung M. B.* The continuous phase heat and mass-transfer properties of dispersions Chem. Eng. Sci. 16 (1961) p. 39–54

14 *Chang S. C.* The method of space-time conservation element and solution element-A new approach for solving the Navier-Stokes and Euler equations J. Comput. Phys. 119 (1995) p. 295–324

15 *Chang S. C.* Courant number insensitive CE/SE schemes, 38th AIAA joint propulsion conference, AIAA-2002-3890, Indianapolis, USA 2002

16 *Chang S. C. Wang X. Y. To W. M.* Application of the space-time conservation element and solution element method to one-dimensional convection-diffusion problems J. Comput. Phys. 165 (2000) p. 189–215

17 *Chang S. C. Wang X. Y. Chow C. Y.* The space-time conservation element and solution element method: A new high-resolution and genuinely multidimensional paradigm for solving conservation laws J. Comput. Phys. 156 (1999) p. 89–136

18 *De Swart S.W.A. Krishna R.* Simulation of the transient and steady-state behavior of a bubble column slurry reactor for Fischer-

Tropsch synthesis Chem. Eng. Processing 41 (2002) p. 35–47

19 *Dorfi E. A. Drury L. O'C.* Simple adaptive grids for 1-D initial value problems J. Comput. Phys. 69 (1987) p. 175–195

20 *Dufort E. C. Frankel S. P.* Stability conditions in the numerical treatment of parabolic differential equations, Mathematical Tables and Other Aids to Computation 7 (1953) p. 135–152

21 *Finlayson B. A.* Nonlinear analysis in chemical engineering, McGraw-Hill, NewYork 1980

22 *Furzeland R. M. Verwer J. G. Zegeling P. A.* A numerical study of three moving-grid methods for one-dimensional partial differential equations which are based on the method of lines J. Comput. Phys. 89 (1990) p. 349–388

23 *Hansen K. Jørgensen S. B.* Dynamic modeling of a gas phase catalytic fixed-bed reactor I-III, Chemical Engineering Science 31 (1976) p. 473–479, 579–598

24 *Harten A. Engquist B. Osher S. Chakravarthy S.* Uniformly high order essentially non-oscillatory schemes III J. Comp. Phy. 71 (1987) p. 231–303

25 *Henson M. A.* Dynamic modeling of microbial cell populations, Current Opinion in Biotechnology 14 (2003) p. 460–467

26 *Heydweiller J. C. Sincovec R. F. Fan L. T.* Dynamic simulation of chemical processes described by distributed and lumped parameter models Comp. Chem. Eng. 1 (1977) p. 125–131

27 *Hill P. J. Ng K. M.* New discretization procedure for the breakage equation AIChE J. 41(5) (1995) p. 1204–1216

28 *Hindmarsch A. C.* LSODE and LSODI: two new initial value ordinary differential equation solvers, ACM SIGNUM newsletter, 15 (1980) p. 19–21

29 *Hoffman J. D.* Numerical methods for engineers and scientists, McGraw-Hill Inc., Mechanical engineering series Part III. Partial differential equations (1993) p. 371–774

30 *Huang W. Russell R. D.* Analysis of moving mesh partial differential equations with spatial smoothing SIAM J. Num. Anal. 34 (1997) 1106–1 p. p. 126

31 *Huang W. Russell R. D.* Moving mesh strategy based on a gradient flow equation for two-dimensional problems SIAM Sci. Comput. 20(3) (1999) p. 998–1015

32 *Hyman J. M. Li S. Petzold L. R.* An adaptive moving mesh method with static rezoning for partial differential equations Computers and Mathematics with Applications 46 (2003) p. 1511–1524

33 *Jiang G. Shu C. W.* Efficient implementation of weighted ENO schemes J. Comp. Phy. 126 (1996) p. 202–228

34 *Kaczmarski K. Mazzotti M. Storti G. Morbidelli M.* Modeling fixed-bed adsorption columns through orthogonal collocations on moving finite elements Comp. Chem. Eng. 21 (1997) p. 641–660

35 *Kohler R. Gerstlauer A. Zeitz M.* Symbolic preprocessing for simulation of PDE models of chemical processes Mathematics and Computers in Simulation 56 (2001) p. 157–170

36 *Krishna R. van Baten J. M.* Eulerian simulations of bubble columns operating at elevated pressures in the churn turbulent flow regime Chemical Engineering Science 56 (2001) p. 6249–6258

37 *Krishna R. van Baten J. M.* Urseanu M. I. Three-phase Eulerian simulations of bubble column reactors operating in the churn-turbulent regime: a scale up strategy Chemical Engineering Science 55 (2000) p. 3275–3286

38 *Kumar S. Ramkrishna D.* On the solution by discretization balance equations by discretization-III. Nucleation, growth and aggregation of particles Chem. Eng. Sci. 52(24) (1997) p. 4659–4679

39 *Lax P. D. Wendroff B.* Systmes of conservation laws Comm. Pure and Appl. Math. 13 (1960) p. 217–237

40 *Leveque R. J.* Finite difference methods for PDEs Lecture note Department of Mathematics University of Washington 1998

41 *Li S.* Adaptive mesh methods and software for time dependent PDEs PhD thesis University of Minnesota 1998

42 *Li S. Petzold L.* Moving mesh methods with upwinding schemes for time-dependent PDEs J. Comput. Phys. 131 (1997) p. 368–377

43 *Lim Y. I. Jørgensen S. B.* A fast and accurate numerical method for solving simulated moving bed (SMB) chromatographic separation problems Chem. Eng. Sci. 59(10) (2004) p. 1931–1947

44 *Lim Y. I. Chang C. S. Jørgensen S. B.* A novel partial differential algebraic equation

(PDAE) solver: iterative conservation element/solution element (CE/SE) method Comput. Chem. Eng. .28(8) (2004) p. 1309–1324

45 *Lim Y. I. Le Lann J. M. Joulia X.* Accuracy, temporal performance and stability comparisons of discretization methods for the solution of Partial Differential Equations (PDEs) in the presence of steep moving fronts Comp. Chem. Eng. 25 (2001a) p. 1483–1492

46 *Lim Y. I. Le Lann J. M. Joulia X.* Moving mesh generation for tracking a shock or steep moving front Comp. Chem. Eng. 25 (2001b) p. 653–663

47 *Lim Y. I. Le Lann J. M. Meyer X. M. Joulia X. Lee G. B. Yoon E. S.* On the solution of Population Balance Equations (PBE) with accurate front tracking methods in practical crystallization processes Chem. Eng. Sci. 57 (2002) p. 177–194

48 *Liu Y. Jacobsen E. W.* On the use of reduced order models in bifurcation analysis of distributed parameter systems Computers and Chemical Engineering 28 (2004) p. 161–169

49 *MacCormark R. W.* The effect of viscosity in hypervelocity impact cratering American Institute of Aeronautics and Astronautics paper (1969) p. 69–354

50 *Mackenzie J. A.* Robertson M. L. The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method J. Compt. Phys. 161(2) (2000) p. 537–557

51 *Mantzaris N. V. Daoutidis P. Srienc F.* Numerical solution of multi-variable cell population balance models: I. Finite difference methods Computers and Chemical Engineering 25 (2001a) p. 1411–1440

52 *Mantzaris N. V. Daoutidis P. Srienc F.* Numerical solution of multi-variable cell population balance models: III. Finite element methods Computers and Chemical Engineering 25 (2001b) p. 1463–1481

53 *Mao G. Petzold L. R.* Efficient integration over discontinuities for differential-algebraic systems Computers and Mathematics with Applications 43 (2002) p. 65–79

54 *Mhaskar P. Hjortsø M. A. Henson M. A.* Cell population modeling and parameter estimation for continuous cultures of *S. cerevisiae* Biotechnol. Prog. 18 (2002) p. 1010–1026

55 *Miller K. Miller R. N.* Moving finite elements I SIAM J. Numer. Anal. 18 (1981) p. 1019–1032

56 *Molls T. Molls F.* Space-time conservation method applied to Saint Venant equations J. Hydraulic Eng. 124(5) (1998) p. 501–508

57 *Motz S. Mitrovic A. Gilles E.-D.* Comparison of numerical methods for the simulation of dispersed phase systems Chem. Eng. Sci. 57 (2002) p. 4329–4344

58 *Oh M.* Modeling and simulation of combined lumped and distributed processes, PhD thesis, University of London 1995

59 *Petzold L. R.* A description of DASSL: A differential/algebraic system solver in scientific computing , eds. R. S. Stepleman et al., North-Holland, Amsterdam (1983) p. 65–68

60 *Poulain C. A. Finlayson B. A.* A Comparison of Numerical Methods Applied to Nonlinear Adsorption Columns" Int. J. Num. Methods Fluids 17(10) (1993) p. 839–859

61 *Powell M. J. D.* On the convergence of the variable metric algorithm J. Inst. Math. Appl. 7 (1971) p. 21–36

62 *Recke B. Jørgensen S. B.* Nonlinear dynamics and control of a recycle fixed bed reactor Proceedings of the 1997 American Control Conference vol 4 (1997) p. 2218–2222

63 *Sargousse A. Le Lann J. M. Joulia X. Jourda L.* DISCo: un nouvel environnement de simulation orient objet; Proceeding of MOSIM 1999 61–66, (1999) p. Nancy, France

64 *Saucez P. Schiesser W. E. van de Wouwer A.* Upwinding in the method of lines Mathematics and Computers in Simulation 56 (2001) p. 171–185

65 *Schiesser W. E.* The numerical method of lines-integration of partial differential equations Academic press New York 1991

66 *Shi J. Hu C. Shu C. W.* A technique of treating negative weights in WENO schemes Brown University Scientific Computing Report Series BrownSC2000 (2000) p. 15 USA

67 *Shu C. W. Osher S.* Efficient implementation of essentially non-oscillatory shock-capturing schemes J. Comp. Phy. 77 (1988) p. 439–471

68 *Shu C. W. Osher S.* Efficient implementation of essentially non-oscillatory shock-capturing schemes II J. Comp. Phy. 83 (1989) p. 32–78

69 *Shu C. W.* Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws ICASE Report No. (1997) p. 97–65

70 *Stockie J. M. Mackenzie J. A. Russell R. D.* A moving mesh method for one-dimensional hyperbolic conservation laws SIAM J. Sci. Comput. 22(5) (2001) p. 1791–1813

71 *Van der Laan G. P. Beenackers A. Krishna R.* Multicomponent reaction engineering model for Fe-catalyzed Fischer-Tropsch synthesis in commercial scale slurry bubble column reactors Chem. Eng. Sci. 54 (1999) p. 5013–5019

72 *Vande Wouwer A. Saucez P. Schiesser W. E.* Some user-oriented comparisons of adaptive grid methods for partial differential equations in one space dimension App. Num. Math. 26 (1998) p. 49–62

73 *Villadsen J. Michelsen M. L.* Solution of differential equation models by polynomial approximation. Prentice-Hall Englewood Cliffs New Jersey 1978

74 *Wey J. S.* Analysis of batch crystallization processes Chem. Eng. Commun. 35 (1985) p. 231–252

75 *Wu J. C. Fan L. T. Erickson L. E.* Three-point backward finite-difference method for solving a system of mixed hyperbolic – parabolic partial differential equations Comp. Chem. Eng. 14 (1990) p. 679–685

76 *Yu S. T. Chang S. C.* Treatment of stiff source terms in conservation laws by the method of space-time CE/SE AIAA 97-0435 35th Aerospace Sciences Meeting, Reno, USA 1997

77 *Zhu G.-Y. Zamamiri A. Henson M. A. Hjortsø M. A.* Model predictive control of continuous yeast bioreactors using cell population balance models Chem. Eng. Sci. 55 (2000) p. 6155–6167