discretization methods the grid is usually locally structured, i.e. each grid node may be considered the origin of a local coordinate system, whose axes coincide with grid lines. This also implies that two grid lines belonging to the same family, say ξ_1, do not intersect, and that any pair of grid lines belonging to different families, say $\xi_1 = \text{const.}$ and $\xi_2 = \text{const.}$, intersect only once. In three dimensions, three grid lines intersect at each node; none of these lines intersect each other at any other point. Figure 3.1 shows examples of one-dimensional (1D) and two-dimensional (2D) Cartesian grids used in FD methods.

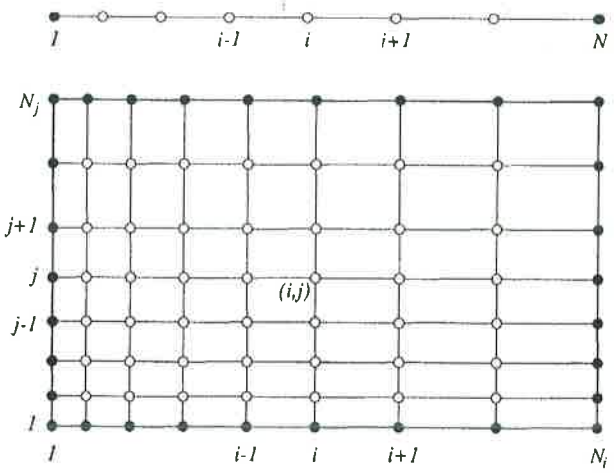

Fig. 3.1. An example of a 1D (above) and 2D (below) Cartesian grid for FD methods (full symbols denote boundary nodes and open symbols denote computational nodes)

Each node is uniquely identified by a set of indices, which are the indices of the grid lines that intersect at it, (i, j) in 2D and (i, j, k) in 3D. The neighbor nodes are defined by increasing or reducing one of the indices by unity.

The generic scalar conservation equation in differential form, (3.1), serves as the starting point for FD methods. As it is linear in ϕ, it will be approximated by a system of linear algebraic equations, in which the variable values at the grid nodes are the unknowns. The solution of this system approximates the solution to the partial differential equation (PDE).

Each node thus has one unknown variable value associated with it and must provide one algebraic equation. The latter is a relation between the variable value at that node and those at some of the neighboring nodes. It is obtained by replacing each term of the PDE at the particular node by a finite-difference approximation. Of course, the numbers of equations and unknowns must be equal. At boundary nodes where variable values are given (Dirichlet conditions), no equation is needed. When the boundary conditions

involve derivatives (as in Neumann conditions), the boundary condition must be discretized to contribute an equation to the set that must be solved.

The idea behind finite difference approximations is borrowed directly from the definition of a derivative:

$$\left(\frac{\partial \phi}{\partial x}\right)_{x_i} = \lim_{\Delta x \to 0} \frac{\phi(x_i + \Delta x) - \phi(x_i)}{\Delta x} . \tag{3.2}$$

A geometrical interpretation is shown in Fig. 3.2 to which we shall refer frequently. The first derivative $\partial\phi/\partial x$ at a point is the slope of the tangent to the curve $\phi(x)$ at that point, the line marked 'exact' in the figure. Its slope can be approximated by the slope of a line passing through two nearby points on the curve. The dotted line shows approximation by a *forward difference*; the derivative at x_i is approximated by the slope of a line passing through the point x_i and another point at $x_i + \Delta x$. The dashed line illustrates approximation by *backward difference*: for which the second point is $x_i - \Delta x$. The line labeled 'central' represents approximation by a *central difference*: it uses the slope of a line passing through two points lying on opposite sides of the point at which the derivative is approximated.

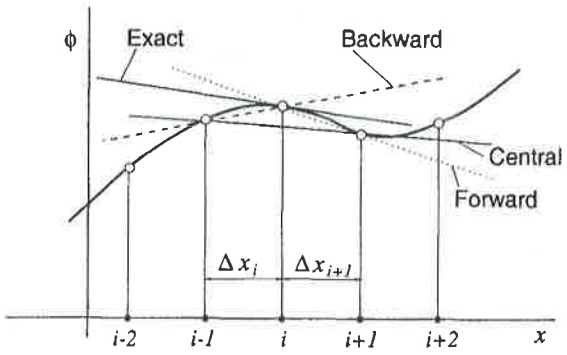

Fig. 3.2. On the definition of a derivative and its approximations

It is obvious from Fig. 3.2 that some approximations are better than others. The line for the central difference approximation has a slope very close to the slope of the exact line; if the function $\phi(x)$ were a second-order polynomial and the points were equally spaced in x-direction, the slopes would match exactly.

It is also obvious from Fig. 3.2 that the quality of the approximation improves when the additional points are close to x_i, i.e. as the grid is refined, the approximation improves. The approximations shown in Fig. 3.2 are a few of many possibilities; the following sections outline the principal approaches to deriving approximations for the first and second derivatives.

$$\phi_1 = \frac{18\phi_2 - 9\phi_3 + 2\phi_4}{11} - \frac{6\,\Delta x}{11}\left(\frac{\partial\phi}{\partial x}\right)_1 . \tag{3.41}$$

Approximations of lower or higher order can be obtained in a similar way.

3.8 The Algebraic Equation System

A finite-difference approximation provides an algebraic equation at each grid node; it contains the variable value at that node as well as values at neighboring nodes. If the differential equation is non-linear, the approximation will contain some non-linear terms. The numerical solution process will then require linearization; methods for solving these equations will be discussed in Chap. 5. For now, we consider only the linear case. The methods described are applicable in the non-linear case as well. For this case, the result of discretization is a system of linear algebraic equations of the form:

$$A_P\phi_P + \sum_l A_l\phi_l = Q_P , \tag{3.42}$$

where P denotes the node at which the partial differential equation is approximated and index l runs over the neighbor nodes involved in finite-difference approximations. The node P and its neighbors form the so-called *computational molecule*; two examples, which result from second and third order approximations, are shown in Fig. 3.4. The coefficients A_l depend on geometrical quantities, fluid properties and, for non-linear equations, the variable values themselves. Q_P contains all the terms which do not contain unknown variable values; it is presumed known.

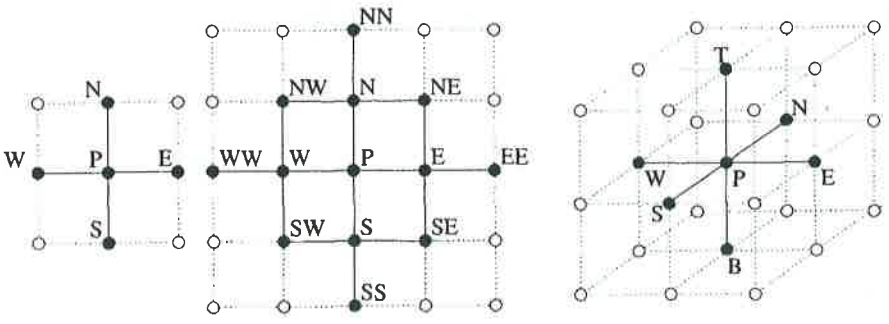

Fig. 3.4. Examples of computational molecules in 2D and 3D

The numbers of equations and unknowns must be equal, i.e., there has to be one equation for each grid node. Thus we have a large set of linear