CHAPTER 5

THREE-DIMENSIONAL NUMERICAL MODELLING

In order to assist in optimising the design of such structures as marine outfalls, physical models have historically been used, scaling down the coastal region of interest, constructing a replica in a large basin and re-scaling all physical parameters in the same manner. Technical constraints in terms of scaling (e.g., horizontal scales much larger than vertical scale, hydraulic scales different from sediment scales), however, lead to one of the main disadvantages of physical models: the level of turbulence (the governing mixing process), chemical and biological processes, and decay rates within the model are likely to differ considerably from those in the real case.

This particular problem does not appear when numerical models are used, although it must be kept in mind that the correct numerical simulation depends on how accurately the modelled equations reflect the real physical conditions in the coastal zone, and also on the type of boundary conditions and closure submodels implemented in the code. Unlike physical models, once the accuracy of the governing equations has been established, numerical models may be used to simulate a large number of different cases, making it a useful tool for civil engineering.

This chapter briefly describes different numerical methods used to solve the transport equations, comparing their main strongpoints and disadvantages, and presenting the reasons for which a Lagrangian numerical approach to the transport problem was selected. A detailed description of the various modules that compose the numerical model developed as the main goal of this thesis follows, and the results from a number of different simple cases, designed to test different features of the model, are finally presented and discussed.

5.1 NUMERICAL METHODS OF SOLUTION

Several numerical techniques have been developed over the years to solve the transport equation both in marine systems (e.g., Holly and Polatera, 1984; Chatwin and Allen, 1985) and other areas (the same equation is applicable, for instance, to the study of flow through porous media -Kumar, 1983-, to the spread of solute in a liquid flowing down a tube -Aris, 1956- or to the impurity redistribution by diffusion and drift during epitaxial growth of semi-conductors -Rohdin *et al.*, 1982). A general classification of these numerical methods divides them into three main categories: Eulerian, Lagrangian, and integral. In contrast with the former two, the integral approach requires an assumption regarding the (spatial) distribution of the relevant variables, as described below.

a) Eulerian methods

Pure Eulerian techniques directly solve the transport equation at a set of spatially fixed grid points. These methods solve the set of algebraic equations that result from the discretisation of the governing partial differential equation, and which contain unknown variables, typically the concentration at a number of fixed locations. The transformation from the initial differential equation to the system of algebraic equations is generally achieved using techniques such as finite differences, Galerkin or Petrov-Galerkin algorithms in finite elements, or collocation. Other techniques used for the transformation are the so-called *boundary element method* (Ikeuchi and Onishi, 1983), and different mixing-cell models (Banks, 1974).

The use of Eulerian methods for transport processes very often require a simultaneous solution of a hyperbolic operator, corresponding to the advective component of transport, and a parabolic operator, which describes the diffusive part. When diffusion dominates, standard Eulerian numerical techniques, whose nature relies on the smoothness of the computing results, can be used to solve the problem. However, when advection is the dominant transport process, these techniques may introduce severe oscillations, resulting in overshoot, undershoot or negative values in the vicinity of high gradients of scalar values (Glass and Rodi, 1982), which can be eliminated using higher-order approximations in space and imposing restrictions on the Péclet number. On the other hand, the stability requirements demanded by explicit methods set restrictions on the Courant number, whereas implicit methods lead to the use of non-symmetric matrices, involving high computational costs in both cases.

b) Lagrangian methods

Lagrangian-type methods solve an analogue of the transport equation (see § 5.2.1) referred to a set of particles, which may be viewed as the nodes of a deformable grid moving with the flow; therefore, an explicit treatment of the hyperbolic ("advective") operator is no longer necessary. However, in some cases, such as when the model is used to simulate transport in regions of rapidly changing fluid velocity, the co-flowing grid becomes severely deformed, limiting the applicability of the model.

By referring the equation analogue to a deformable coordinate system, the transport problem is reduced to finding the solution to a diffusion equation on a new variable grid at each timestep. The numerical treatment of this equation may be done by finite element methods (Sobey, 1982), finite difference methods specially designed for non-uniform grids (McBride and Rutherford, 1984), or random walk methods (Hunter, 1987).

Zannetti (1990) differentiates between four different types of Lagrangian models depending on the method used to solve the transport equation: Lagrangian box -or trajectory- models, Gaussian segmented plume models, Gaussian puff model and particle models.

c) Integral methods

Also called Gaussian methods, they are characterised by solving the ordinary differential equations along characteristics of the flow problem. The typical example is the modelling of a jet discharge, where the jet centreline is the characteristic (see, for instance, Chu & Lee, 1996); by relating all the problem conditions to the characteristic, the overall behaviour may be calculated by a simple forward integration of the ordinary differential equations. The conditions elsewhere are then found by integrating the forces, fluxes, etc., over the jet cross-section, based on a pre-selected (generally Gaussian) cross-section distribution.

It can be seen from experimental studies, however, that average concentration values given by this type of models can be smaller than real values by as much as a factor of 6. This is because the actual mixing process which creates dilution is defined by the smaller energy-dissipating eddies, whereas integral methods perform long-term averages, which correspond to the larger energy-containing eddies that push around coherent volumes of fluid. Put in another way, the time-average performed by integral models corresponds to the transport induced by large eddies, which simply move "fluid packets" around, while it ignores small-scale eddies. Integral models therefore average groups of "packets", including regions of large concentration (the "packets") and regions of very low concentration values, yielding concentrations lower than experimental values. List and Dugan (1992) have pointed out, consequently, that integral models "describe averaged flow conditions that never really exist as physical entities. The flows described are actually artefacts of the averaging process and the modelling procedures".

The Eulerian and Lagrangian approaches are sometimes merged into Eulerian-Lagrangian (ELA) grid methods, in which the transport equation is decoupled into two parts using a time-splitting process (Noye, 1986). The advective component is solved using one of the Lagrangian techniques, and the results are interpolated onto an Eulerian grid, on which a finite difference technique is then used to solve the diffusion part of the equation. By combining both methods, the efficiency of the Eulerian treatment of the diffusion term is coupled with the accuracy of the Lagrangian approach to the advective part of the transport equation.

A further combination of the aforementioned methods is the hybrid Eulerian-Lagrangian / Random Walk model (Moeller and Adams, 1994), which uses particles to represent the near field, and relies on the Eulerian-Lagrangian approach to model the far field. This type of model becomes useful when the number of particles necessary to model the transport process with the required accuracy is prohibitive due to computational costs.

5.1.1 Choosing the model

The type of model to be developed or used in each case must be chosen after performing a thorough evaluation of the specific problem to be solved. This evaluation should consider, amongst others;

- The number, and type, of relevant processes that are to be modelled (physical and non-physical).
- The spatial resolution that will be required for the solution.
- The existence, or possible existence, of unsteady conditions.
- The computational expenses, and the available computer capacity.

Depending on the case to which the model is going to be applied, one numerical formulation may be better suited than another due to its particular characteristics. A list of advantages and disadvantages of each approach is given below, compiled from a series of authors (Hansen, 1991; Hunter, 1987; Zannetti, 1990; O'Connor & Nicholson, 1988; Spaulding & Pavish, 1984; Moeller & Adams, 1994; Nguyen *et al.*, 1997):

a) Eulerian approach

- Solving two different operators simultaneously may be difficult in cases where the total transport is dominated by advective mechanisms. This shortcoming may be bypassed with the use of high-order methods or time-splitting techniques.

- When the diffusion processes are scale-dependent, Eulerian models may present numerical problems.

- Since all the modelled properties are uniformly distributed throughout the computational cells, Eulerian models show strong artificial diffusion, which tends to smooth out spatial gradients.

- It presents difficulties when modelling effluents of various particle sizes and density characteristics.

- It produces severely depressed concentration gradients upstream of point source waste loads.

- It requires the input of artificial stability classes, which determine the computational timestep.

- The conservation of pollutant mass has to be continuously checked.

- The effect of volumetric forces, like pollutant buoyancy, is easily incorporated into the model formulation.

- However, it can be hard to simulate buoyancy effects due to the need of fine vertical resolution, which can collide with hardware restrictions.

- It can (easily) incorporate second- and higher-order equations to describe chemical kinetics.

b) Lagrangian approach

- Models based on a Lagrangian scheme are practically free of restricting physical assumptions, since all uncertainties can be combined into the determination of the pseudovelocities that appear in the transport equation analogue. Their programming structure is preferable for non-homogeneous flows and parallel computing.

- Each particle can be tagged with its characteristics (coordinates, source indicator, density, etc), therefore allowing the simulation of multi-component effluents. This is particularly useful when transport and fate processes are best described by attributes of the individual particles.

- The accuracy of the diffusion modelling is reduced because of the use of stochastic methods. However, numerical diffusion is non-existent until the particles are projected onto a numerical grid.

- The computational timestep is solely determined by physical criteria, since there are no stability requirements to be met.

- The conservation of mass is implicit within the computational scheme.

- It is not easy to include volumetric forces, such as buoyant spreading, due to the discrete nature of the Lagrangian elements.

- Concentration values obtained from particle density distributions fluctuate with the number of particles used. Since the statistical fluctuations are proportional to $N^{1/3}$, an increase in the number of particles does not improve concentration to the same degree.

- It can be very CPU-time consuming.

- It is useful to simulate the transport and spreading of patches of the same size as, or smaller than, the hydrodynamic mesh size.

c) Integral approach

- Integral models are comparatively very fast to run.

- It is restricting when the characteristics are unknown, or when the cross-section distribution may smear out important features of the structure.

- It shows no problems of space resolution, since the solution is given along a characteristic line, and the spatial separation is determined by the timestep employed.

- It is generally used for –and limited to– slow changing conditions (semi stationary situations).

Although each type of formulation is adequate for a specific group of problems, several authors have performed direct comparisons between different models in order to determine which approach is preferable under given conditions. Hunter (1987), for instance, compared a Lagrangian and a Eulerian model, and concluded that a particle-based formulation was more efficient, in a computational sense, for cases of higher dimension (3D transport problems, rather than 2D), and when the pollutant patch occupies only a small portion of the total computational domain; this is so because calculations in Eulerian models extend over the whole domain, whereas Lagrangian algorithms concentrate their calculations in the region occupied by particles. In addition, his treatment of the motion of surface oil patches revealed that the compared efficiency of Lagrangian models was greater when working with a relatively small amount of particles, although better accuracy is obtained when the number of particles is large.

On the other hand, Moeller and Adams (1994) compared three different models -namely a random walk approach, a ELA, and a hybrid model- in a set of different flow fields, and observed that, for two-dimensional pollutant transport, the hybrid approach (random walk in the nearfield, and ELA in the farfield) was the most accurate for shear flows, mainly due to its greater capability to model the nearfield with respect to ELA, whilst the pure particle model performed the worst of all.

The goal of this thesis is to develop a fully-3D transport model to be applied in coastal waters. Therefore, the numerical model must be able to accurately reproduce horizontal and vertical advection and mixing. Moreover, and because the main transport processes in coastal regions are related either to sediment loads or wastewater discharges, the model must also be able to distinguish different types of particles (e.g., sediment grains of different sizes and densities) and treat them in a separate manner, including different characteristic transport mechanisms for each case (resuspension and settling for sediments, buoyancy effects and microbiological decay for wastewater discharges). Taking into account the desired model features outlined above, and the different types of numerical models and their characteristics, it has been decided to follow the Lagrangian random walk approach to develop the present numerical code.

5.2 THE LAGRANGIAN PARTICLE MODEL

Particle-tracking based methods have been extensively used over the years in order to model one-, two-, and three- dimensional solute transport in groundwater (e.g., Tompson and Gelhar, 1990), but their application to transport in surface waters has been rather sparse until recent years (Dimou and Adams, 1990). The main advantage for using particle tracking models in groundwater problems is the frequent occurrence of small-scale variabilities that can be modelled without grid resolution expense; in surface waters, the main motivation appears to be the easy simulation of advection-dominated transport problems. Moreover, particle models have been used in a wide spectrum of applications ranging from the atomic to the astronomical scale (polytropic stellar models -Gingold and Monaghan, 1977), with other important applications to plasma and turbulent fluid dynamics. In coastal and oceanic contexts, particle tracking methods have been used to model sea ice transport (Flato, 1993), oceanic fronts (Pavia and Cushmanroisin, 1988), oil spills (Proctor *et al.*, 1994), tidal dispersion processes (Allen, 1982; Van Dam, 1994), and chaotic stirring in a tidal environment (Heemink, 1990; Geyer, 1993), amongst others.

These models use a certain number of computational (fictitious) particles to simulate the dynamics of a selected parameter, such as mass or heat. The motion of the particles can be produced both by deterministic velocities and random pseudovelocities generated by Monte-Carlo techniques. In the latter case, the trajectory of each particle must be considered as a single realisation of an infinite set of possible solutions. However, if enough particles are used, the average properties of the particle ensemble are not affected by the randomness of the pseudovelocities.

In general, the relation between the computational particles and the physical (real) particles, which is an important factor for the interpretation of the simulation results, fall into one of the following three categories (Zannetti, 1990):

- a one-to-one correspondence between actual and simulated particles, as is the case of molecular dynamics.

- a description of fluid elements as particles; here the correspondence between the physical particle and the numerical particle is lost.

- the use of "superparticles" which represent a cloud of physical particles with similar characteristics (e.g., dynamics of galaxies).

In the present model, the Lagrangian particles correspond to elements of fluid, and each particle is tagged with the properties of the fluid (e.g., density, pollutant concentration, suspended sediment concentration, etc.).

5.2.1 The transport equation, and the equation analogue

Consider -figure 5.1- a small cubic volume element $\Delta V = \Delta x \Delta y \Delta z$, located at a fixed point within a fluid moving with a velocity (u,v,w). The amount of a contaminant substance transported by the fluid entering ΔV in each direction during a timestep Δt is:

$$\begin{aligned}
\Delta m_{yz} &= -\left[\frac{\partial}{\partial x}(uC) + \frac{\partial}{\partial x}\left(-D_x \frac{\partial C}{\partial x}\right)\right]\Delta x \Delta y \Delta z \Delta t \\
\Delta m_{xz} &= -\left[\frac{\partial}{\partial y}(vC) + \frac{\partial}{\partial y}\left(-D_y \frac{\partial C}{\partial y}\right)\right]\Delta x \Delta y \Delta z \Delta t \\
\Delta m_{xy} &= -\left[\frac{\partial}{\partial z}(wC) + \frac{\partial}{\partial z}\left(-D_z \frac{\partial C}{\partial z}\right)\right]\Delta x \Delta y \Delta z \Delta t
\end{aligned} \tag{5.1}$$

where $\boldsymbol{D} = (D_x, D_y, D_z)$ is a vector of molecular diffusivity, C is the concentration of the substance, and m_{ij} represents the mass passing through a surface defined by the axes i-j.

The variation of pollutant mass inside ΔV is then

$$\begin{aligned}
\frac{\Delta m}{\Delta V \Delta t} &= -\left[\frac{\partial}{\partial x}(uC) + \frac{\partial}{\partial y}(vC) + \frac{\partial}{\partial z}(wC)\right] + \\
&+ \left[\frac{\partial}{\partial x}\left(D_x \frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_y \frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(D_z \frac{\partial C}{\partial z}\right)\right] + Q_{SC}
\end{aligned} \tag{5.2}$$

where Q_{SC} is a term accounting for the possible presence of sources and sinks in ΔV. By making Δt tend to zero, the final form of the instantaneous transport equation for laminar flows is obtained:

$$\frac{\partial C}{\partial t} + u\nabla C = \nabla(D\nabla C) + Q_{SC} \tag{5.3}$$

When the flow is turbulent, all the hydrodynamic and thermodynamic properties undergo chaotic fluctuations due to the eddies within the fluid. Since the estimation of instantaneous concentration values is very expensive under these conditions, it becomes of interest to calculate mean values, averaging in one way or another. The usual approach is to assume that the instantaneous velocity field can be written as $u = \overline{U} + u'$, where u' is a fluctuation contribution added to a mean component $\overline{U}$, and the same goes for the concentration $C = \overline{C} + c'$. Substituting these expressions in equation (5.3) yields,

$$\frac{\partial \overline{C}}{\partial t} + \overline{U}\frac{\partial \overline{C}}{\partial x} + \overline{V}\frac{\partial \overline{C}}{\partial y} + \overline{W}\frac{\partial \overline{C}}{\partial z} = D\nabla^2 \overline{C} - \frac{\partial(\overline{u'c'})}{\partial x} - \frac{\partial(\overline{v'c'})}{\partial y} - \frac{\partial(\overline{w'c'})}{\partial z} + Q_{SC} \tag{5.4}$$

where ***D*** has been assumed constant and isotropic, and the following averaging rules have been used:

$$\overline{F+G} = \overline{F} + \overline{G}$$

$$\overline{cF} = c\overline{F} \qquad c = \text{constant}$$

$$\overline{\overline{F}G} = \overline{F} \cdot \overline{G} \tag{5.5}$$

$$\overline{F'} = 0$$

$$\frac{\overline{\partial F}}{\partial x} = \frac{\partial \overline{F}}{\partial x}$$

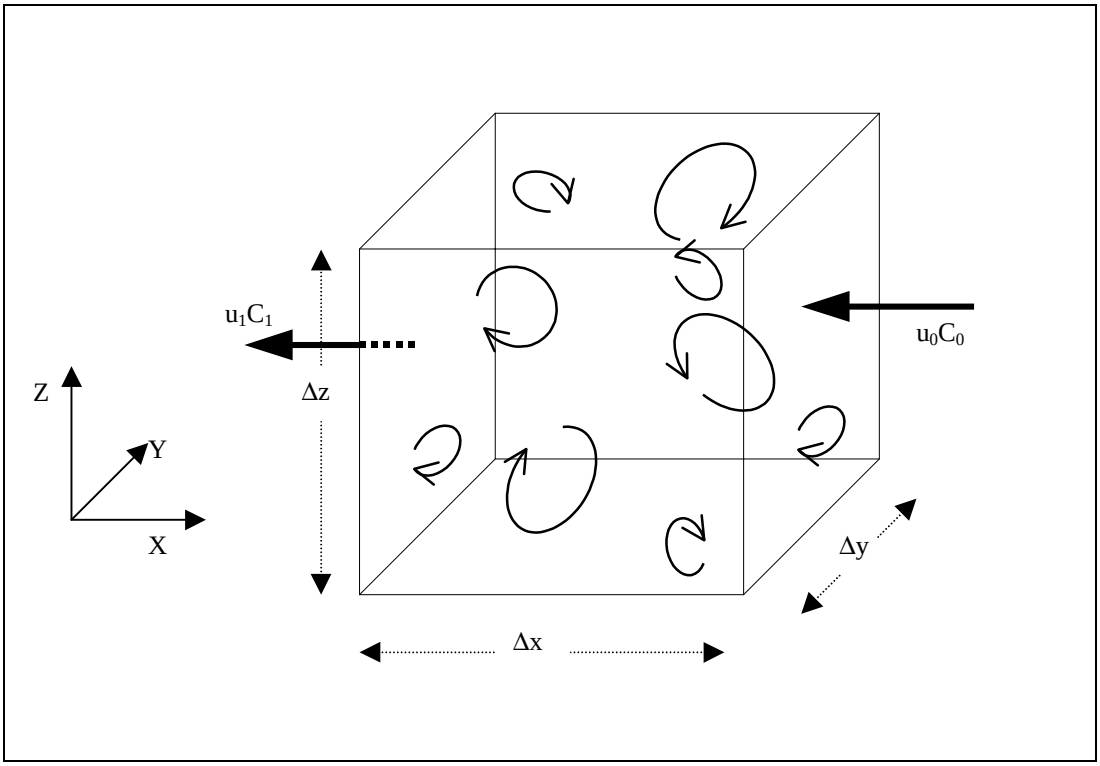

Figure 5.1: Scheme for the derivation of the transport equation.

The quantities $\partial\left(\overline{u_i'c'}\right)/\partial x_i$ in equation (5.4) describe the contribution of fluctuating components of the velocity to the pollutant balance. By analogy to molecular diffusion, these turbulent fluxes are generally expressed as

$$\overline{u_i'c'} = -K_{ij}\frac{\partial \overline{C}}{\partial x_j} \tag{5.6}$$

with K_{ij} the turbulent diffusion second-order tensor. The general form of (5.4) then becomes

$$\frac{\partial \overline{C}}{\partial t} + \overline{U_i}\frac{\partial \overline{C}}{\partial x_i} = D\nabla^2\overline{C} + \frac{\partial}{\partial x_i}\left(K_{ij}\frac{\partial \overline{C}}{\partial x_j}\right) + Q_{SC} \tag{5.7}$$

Equation (5.7) is the fundamental mathematical statement of the physical phenomenon of transport but, due to its complexity, is seldom solved in an exact manner. Instead, in numerical modelling, approximate solutions are found by using equations that are analogue to this basic transport equation.

In the random walk particle approach to transport problems, the analogue expression to be considered is the Fokker-Planck equation, which describes the time evolution of Markovian processes:

$$\frac{\partial f}{\partial t} + \nabla(\mathbf{A}f) - \nabla^2\left(\frac{1}{2}\mathbf{B}\mathbf{B}^{\mathbf{T}}f\right) = 0 \tag{5.8}$$

where f is a probability density function -$f(\boldsymbol{x}, t_o) = \delta(\boldsymbol{x}\text{-}\boldsymbol{X})$-, $\boldsymbol{A}$ is a deterministic forcing vector acting on each individual particle, and $\boldsymbol{B}$ is a deterministic scaling matrix.

For large enough times, and a finite number of particles, equation (5.8) can be discretised so as to yield the position of a single particle at time t_n, given $\boldsymbol{A}$, $\boldsymbol{B}$ and its position at time t_{n-1} (Tompson and Gelhar, 1990):

$$\mathbf{X}^n = \mathbf{X}^{n-1} + \mathbf{A}(\mathbf{X}^{n-1}, t_{n-1}) + \sqrt{\Delta t}\,\mathbf{Z}\,\mathbf{B}(\mathbf{X}^{n-1}, t_{n-1}) \tag{5.9}$$

with $\boldsymbol{Z}$ a vector containing 3 independent random numbers, with mean zero and variance one, and $\Delta t = t_n - t_{n-1}$. This expression is a discrete representation of the more general Langevin equation

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}(\mathbf{X}, t) + \mathbf{B}(\mathbf{X}, t)\cdot\xi(t) \tag{5.10}$$

which was used originally to describe Brownian motion.

If a number N of particles are moved simultaneously according to equation (5.9), the density of particles lying in a small volume V_g, centred at $\boldsymbol{x}$ and time t can be estimated by

$$f(\mathbf{x}, t) \approx \frac{N_{pg}}{NV_g} \tag{5.11}$$

so that

$$C(\mathbf{x}, t) \approx mNf(\mathbf{x}, t) \tag{5.12}$$

is a dimensional measure of concentration, being m the (constant) mass of each particle. When the number of particles is very large ($N \to \infty$), equation (5.12) becomes an exact solution. The transport equation (5.7) is readily derived from the Fokker-Planck equation (5.8) by choosing $\boldsymbol{A}$ and $\boldsymbol{B}$ as

$$A_i(\mathbf{X}^n, t_n) = u_i(\mathbf{X}^n, t_n) + \frac{\partial(D + K_{ij})}{\partial x_j} \tag{5.13}$$

$$B_{ik} B_{jk} = 2(D + K_{ij}) \tag{5.14}$$

By considering all the physical mechanisms that contribute to the overall transport, and re-writing equation (5.9), the equation analogue that is solved by the current numerical model may be expressed as (Sánchez-Arcilla *et al.*, 1998)

$$\left.\begin{aligned} x^n &= x^{n-1} + (u_a + u_{Dm} + u_{Dx})\Delta t \\ y^n &= y^{n-1} + (v_a + v_{Dm} + v_{Dy})\Delta t \\ z^n &= z^{n-1} + (w_a + w_{Dm} + w_{Dz} + w_S + w_R + w_B)\Delta t \end{aligned}\right\} \tag{5.15}$$

where (u_a,v_a,w_a) are the components of a deterministic advective velocity, w_S, w_R, and w_B are deterministic vertical velocities due to gravitational settling, turbulent resuspension and buoyancy, respectively, and (u_{Dm},v_{Dm},w_{Dm}) and (u_{Dx},v_{Dy},w_{Dz}) are components of random velocities due to molecular and turbulent diffusion, respectively. The randomness of the procedure guarantees that any two particles, initially located at the same point $(\boldsymbol{x_0}, t_0)$, will end up at different locations $(\boldsymbol{x_1}, t_1)$ and $(\boldsymbol{x_2}, t_2)$ after repeated application of the equation set (5.15), therefore allowing particle dispersion.

The velocity components appearing in (5.15), and the methods followed to compute them, are described in the following sections. Additionally, the number of particles in the simulation may increase or decrease if some type of growth or decay is involved; the case of microbiological inactivation is included in the model, and is also described in §5.2.1.6. Finally, §5.2.1.7 introduces the boundary conditions that apply to equations (5.15), and deals with the procedure followed to implement them in the numerical code formulation.

5.2.1.1 Advection term

The advective component of the velocity in equations (5.15) is calculated as the sum of contributions due to the hydrodynamic currents and the wave-induced mass transport. Although it is not strictly an advective term, an additional contribution due to the initial discharge velocity is also included here.

a) Hydrodynamic currents

The first contribution to advection is computed using a velocity field given by an external hydrodynamic model, which also defines the computational grid variables (such as the number and size of grid cells).

The hydrodynamic velocity affecting each individual particle is found by performing a horizontal bilinear interpolation (figure 5.2), and a vertical linear interpolation when the hydrodynamic data is 3D, from the computational grid nodes nearest to the particle's position:

$$F = F_1 + (F_2 - F_1)\frac{y}{\Delta Y} + (F_4 - F_1)\frac{x}{\Delta X} + (F_1 - F_2 + F_3 - F_4)\frac{x}{\Delta X}\frac{y}{\Delta Y} \tag{5.16}$$

where F_i are the values at each node, (x,y) is the distance from the first node to the particle, and ΔX and ΔY are the grid sizes.

When the hydrodynamic velocity field given by the external model is depth-integrated, i.e., the horizontal components are averaged over the whole water column, the transport model can compute vertical profiles induced by wind stresses on the water surface, in order to obtain a three-dimensional flow pattern. For this purpose, the wind-induced circulation model proposed by Massel (1989) – equations 2.25a,b,c – is implemented in the code, and applied at the position of each particle after the mean horizontal components have been obtained.

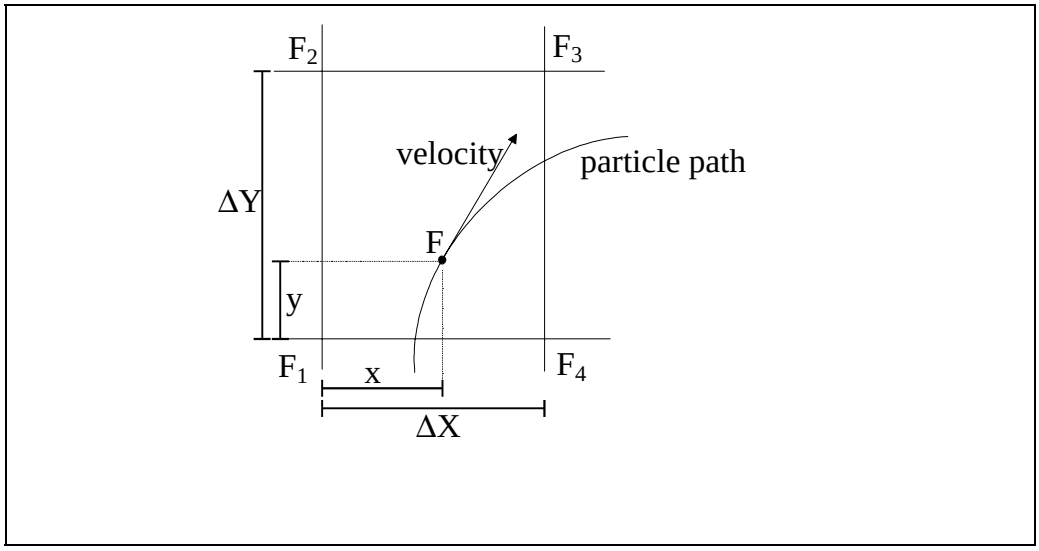

Figure 5.2: Bilinear interpolation scheme for the computation of the advective velocity component.

b) Wave-induced mass transport

An additional component is introduced to account for mass flux induced by oscillatory motion. As mentioned in §2.3.1.3, the effects of wave motion on pollutant transport are twofold: it enhances net advective transport because of mass flow, and increases the level of turbulence. The latter effect is accounted for in the computation of the turbulent term (§5.2.1.2). The former is described by (Davydov, 1989)

$$q_j = \delta_j \frac{\partial C}{\partial x_3} \qquad q_3 = -\delta_j \frac{\partial C}{\partial x_j} \qquad j = 1,2 \tag{5.17}$$

where

$$\delta_j = \frac{\sigma a_w^2 \sinh\left(4\pi \, z/L_w\right)}{4\sinh^2\left(2\pi \, d/L_w\right)} \frac{k_j}{|\mathbf{k}|} \tag{5.18}$$

and σ, a_w, k, and L_w are all wave parameters defined in Chapter 2, and z is the vertical position of the particle.

To compute the wave-induced flow, an auxiliary 3D grid is built in the region occupied by the Lagrangian particles, and the number of particles within each cell is counted. The three components of the "mass flow velocity" are then calculated using the following discretisation of equation (5.17), where the fact that q_j is dimensionally equal to the product of concentration and velocity has been used:

$$q_i(l,m,n) = u_{q_i}(l,m,n)C(l,m,n) \tag{5.19}$$

$$\begin{cases} u_q(l,m,n) = \dfrac{\delta_1}{2\Delta z}\dfrac{N(l,m,n+1)-N(l,m,n-1)}{N(l,m,n)} \\ v_q(l,m,n) = \dfrac{\delta_2}{2\Delta z}\dfrac{N(l,m,n+1)-N(l,m,n-1)}{N(l,m,n)} \\ w_q(l,m,n) = -\dfrac{\delta_1}{2\Delta x}\dfrac{N(l+1,m,n)-N(l-1,m,n)}{N(l,m,n)} - \\ \qquad\qquad -\dfrac{\delta_2}{2\Delta y}\dfrac{N(l,m+1,n)-N(l,m-1,n)}{N(l,m,n)} \end{cases} \tag{5.20}$$

with $N(l,m,n)$ is the number of particles found in the cell of indexes (l,m,n), and Δx, Δy, and Δz are the size of each cell.

c) Initial discharge velocity

An "initial jet" velocity component, computed as a function of the particle distance to the source, a characteristic lengthscale, and a combination of the discharge velocity and the ambient current component parallel to the discharge direction, is assigned to each particle in the simulation. The characteristic lengthscale is taken to be l_m,

$$l_m = \frac{M^{1/2}}{u_a} \tag{5.21}$$

with

$$M = w_0 Q_0 \tag{5.22}$$

where w_0 is the initial discharge velocity, and Q_0 is the discharge flow rate.

At a distance l_m from the source, the trajectory of the discharged jet starts to deflect due to the transverse component of the current. Roberts *et al.* (1989) found that the value of l_m was not affected if the parallel component of the current was neglected, whereas only small errors in the terminal height and the initial dilution appeared. Therefore, the ambient velocity is decomposed into a component parallel to the direction of discharge, and one transverse to this direction. Whereas the former will contribute to the duration of the jet phase, depending upon the relative magnitude of the discharge velocity and the ambient current, the latter contributes to the jet deflection.

Under this scheme, a "parallel momentum flux" is defined as

$$M_{0p} = Q_0\left(w_0 + u_a\cos\sigma\right) \tag{5.23}$$

where w_0 is the discharge velocity (Q_0/A), u_a is the ambient current, and σ is the angle defined by the ambient current and the discharge velocity. The value of a lengthscale l_{mp} can be then computed as

$$l_{mp} = \frac{M_{0p}^{1/2}}{u_a} \tag{5.24}$$

and the jet velocity of each particle at any timestep is calculated as

$$u_j^j = \begin{cases} 1 - a_1 w_0^j & r < 0.8 l_{mp} \\ a_2 w_0^j & r > 0.8 l_{mp} \end{cases} \qquad j = 1,2,3 \tag{5.25}$$

in which the values of the coefficients a_1 and a_2 have been estimated by fitting computed particle trajectories for vertical pure jets -figure 5.3- and for jets ejected at an angle to the flow –figure 5.4- with trajectory data from Chu and Goldberg (1974) –obtaining

$$a_1 = \frac{1.25 r}{l_{mp}}\left(1 - \frac{1.25}{l_{mp}}\right) \quad ; \quad a_2 = \frac{1}{r} \tag{5.26}$$

and where r is the distance of the particle to the source.

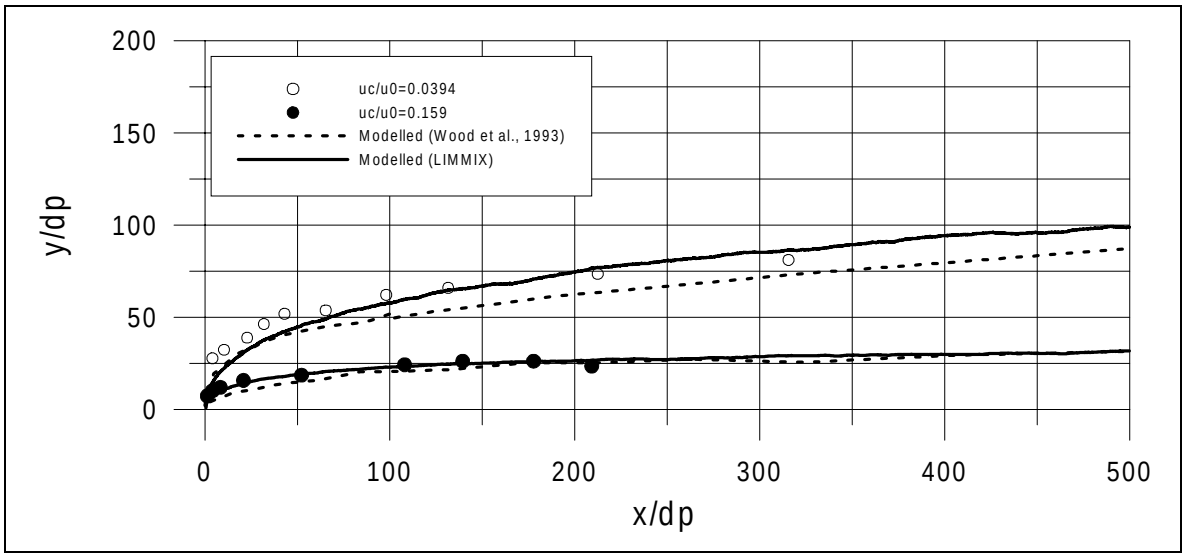

Figure 5.3: Comparison of the computed results with the trajectory data of Chu and Goldberg (1974). *uc* and *u0* represent the discharge and ambient current velocity, respectively.

The total advective velocity is then the sum of the component given by the external hydrodynamic model, modified by the wind stresses if necessary, plus the possible advection due to wave-induced mass flow, and the initial velocity contribution.

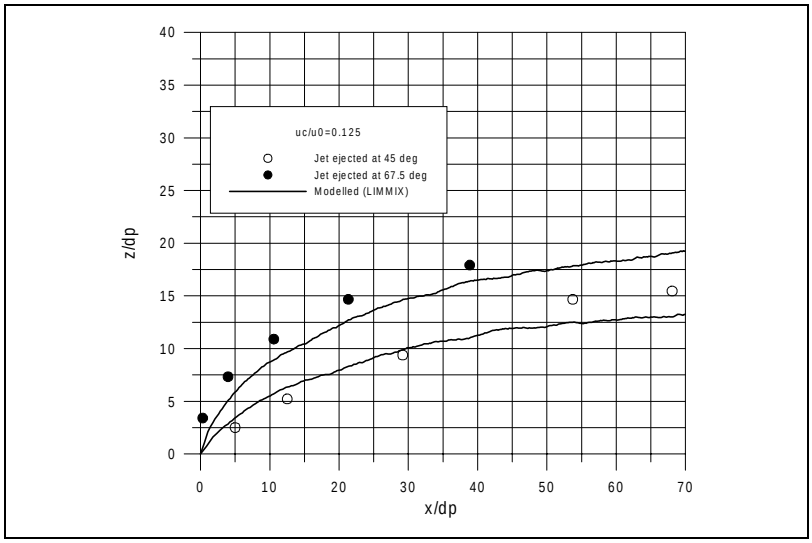

Figure 5.4: Comparison of the computed results with the trajectory data of Chu and Goldberg (1974). *uc* and *u0* represent the discharge and ambient current velocity, respectively.

5.2.1.2 Molecular and turbulent diffusion

The transport of a pollutant by diffusive processes is mainly due to the turbulent nature of the motion of the surrounding fluid, and is determined by the set of turbulent diffusion coefficients that appear in the convection-diffusion equation (5.7). The selection or computation of these parameters often requires the solution to a complicated system of coupled equations which includes a large number of constants and empirical functions that are not always well known. An example is the 14-equation system suggested by Ozmidov (1990):

$$\frac{\partial u}{\partial t} - \Omega v = \frac{\partial}{\partial z}\left(K_V \frac{\partial u}{\partial z} \right) \tag{5.27a}$$

$$\frac{\partial v}{\partial t} + \Omega u = \frac{\partial}{\partial z}\left(K_V \frac{\partial v}{\partial z} \right) \tag{5.27b}$$

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(K_V^T \frac{\partial T}{\partial z} \right) - u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y} \tag{5.27c}$$

$$\frac{\partial S}{\partial t} = \frac{\partial}{\partial z}\left(K_V^S \frac{\partial S}{\partial z} \right) - u\frac{\partial S}{\partial x} - v\frac{\partial S}{\partial y} \tag{5.27d}$$

$$\rho = \rho_0 \left[1 + \alpha_T (T - 27) + \beta_S (S - 36) \right] \tag{5.28}$$

$$\frac{\partial k}{\partial t} = K_V \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right] + \frac{\partial}{\partial z} \left(K_V^k \frac{\partial k}{\partial z} \right) - B - \varepsilon \qquad (5.29a)$$

$$\frac{\partial \theta}{\partial t} = K_V^T \left(\frac{\partial T}{\partial z} \right)^2 + \frac{\partial}{\partial z} \left(K_V^\theta \frac{\partial \theta}{\partial z} \right) - \varepsilon_T \qquad (5.29b)$$

$$\left.\begin{array}{ll} K_z = l\sqrt{k} & \varepsilon_T = \lambda(Ri) 6\chi \left(\frac{\partial T}{\partial z} \right)^2 \\ \varepsilon = a_\varepsilon k^{3/2} / l & \\ l^2 = k / \left[\varphi(Ri) N^2 \right] & Ri = N^2 / \left[(\partial u / \partial z)^2 + (\partial v / \partial z)^2 \right] \\ k = g\alpha_T K_V^T (\partial T / \partial z) + g\beta_S K_V^S (\partial S / \partial z) & N^2 = g / \rho (\partial \rho / \partial z) \end{array}\right\} \qquad (5.30)$$

where (5.27a-b) are the dynamic equations for the horizontal components of the velocity, and (5.27c-d) are the transport equations for heat and salinity, respectively; (5.28) is a state equation for the ambient water, assumed a linear function of temperature and salinity, and (5.29a-b) express the turbulent energy (k) balance, and the balance for temperature inhomogeneities (θ). In these equations, Ω stands for the Coriolis parameter, K_z is the vertical turbulent viscosity coefficient, K_V^T, K_V^S, K_V^k, and K_V^θ are the vertical turbulent diffusivities for heat, salt, turbulent energy, and temperature inhomogeneities, respectively; α_T and β_S are the thermal and salinity coefficients of sea water expansion, ε is the rate of turbulent energy dissipation, ε_T is the rate of dissipation of temperature inhomogeneities, and B is the buoyancy flux. Other parameters are the turbulent scale l, the Richardson number Ri, the Brunt-Väisälä frequency N, the molecular thermal diffusivity χ, and $\varphi(Ri)$, $\lambda(Ri)$ and α_ε, which are some functions of Ri. After defining these functions and adopting values for the physical constants, the system is closed with a total of 19 unknown variables.

Furthermore, the use of implicit schemes or analytical methods to solve the diffusion equation leads to non-zero concentration values in the whole computational domain immediately after the injection of the substance. On the other hand, using an explicit approach clearly defines the pollutant-affected region, but its boundary and the boundary's spatial velocity become very dependent on the timestep and grid spacing of the model. All these inconveniences may be overcome by using direct numerical methods to analyse the motion of discrete particles in turbulent flows.

a) The random walk algorithm

The simulation of diffusive processes by means of the random walk algorithm is based on the assumption that the trajectories of all particles are characterised by being Brownian in nature, that is, the displacement of a particle at a time t is independent of its displacement at any previous time t'. It can be thought of as if each particle position was periodically perturbed by a random vector obtained from a distribution function P; the variance of this distribution is related to the effective diffusion coefficients, as will be seen further on.

So, at the beginning of the random walk process, each individual particle represents a delta function of concentration. After one timestep, the particle has been "spread out", since its new position is determined by a probability distribution that spans a finite volume of space. This process can be conveniently described in terms of the moments of the concentration distribution with respect to the

particle position at the beginning of the timestep; the distribution function should be chosen so that the moments of *P* reflect the change of moments derived from the transport equation.

Using statistical analysis it can be shown that the root-mean-square distance travelled by a particle during a timestep Δt, in a one-dimensional space, is

$$d_{rms} = \sqrt{2K\Delta t} \tag{5.31}$$

For an individual particle, the diffusion step size may be generated randomly by

$$r = [R]_0^s \tag{5.32}$$

where $[R]_0^s$ is a random number in the interval 0 to *s*. The standard deviation of the set of all numbers – between 0 and 1– generated by the typical random number generator found on computer systems is

$$\left[\int_0^1 R^2 dr\right]^{1/2} = \frac{1}{\sqrt{3}} \tag{5.33}$$

so that diffusion steps are then generated by

$$r = 2\left(0.5 - [R]_0^1\right)\sqrt{6K\Delta t} \tag{5.34}$$

where $2\left(0.5 - [R]_0^1\right)$ substitutes $[R]_0^1$ in order to account for both negative and positive displacements. From this equation, the diffusive velocity can be defined as

$$u_D = 2\left(0.5 - [R]_0^1\right)\sqrt{\frac{6K}{\Delta t}} \tag{5.35}$$

which is the diffusion component of the velocity in the set of equations (5.15).

a.1) The choice of the distribution for the random step

As was mentioned before, the distribution function *P* must be chosen so that its moments of any order reflect the variations of the moments of the transport equation. Therefore, some limitations are imposed on the possible forms of *P*.

Consider, for the sake of simplicity, a one-dimensional pure diffusive (i.e., without advection) process with constant diffusivity *K* (the extension to higher dimensions is straightforward). The transport equation, in this case, is

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial C}{\partial x}\right) \tag{5.36}$$

and the moments of order *n* of the concentration distribution can be defined as

$$M_n = \int_{-\infty}^{\infty} x^n C dx \quad n \geq 0 \tag{5.37}$$

or, after normalising,

$$N_n = \frac{M_n}{M_0} \tag{5.38}$$

Performing the time-derivative of equation (5.37), and using (5.36) multiplied by x^n, yields

$$\frac{dM_n}{dt} = \int_{-\infty}^{\infty} x^n \frac{\partial}{\partial x}\left(K \frac{\partial C}{\partial x} \right) dx \tag{5.39}$$

which, after integrating twice by parts gives

$$\frac{dM_n}{dt} = -\int_{-\infty}^{\infty} n x^{n-1} K \frac{\partial C}{\partial x} dx \qquad \text{(first integration)} \tag{5.40}$$

$$\frac{dM_n}{dt} = \int_{-\infty}^{\infty} Kn(n-1) C x^{n-2} dx = Kn(n-1) M_{n-2} \qquad \text{(second integration)} \tag{5.41}$$

where it has been assumed that the concentration C is spatially limited, that is,

$$\begin{cases} x^n \dfrac{\partial C}{\partial x} \to 0 \\ \qquad\qquad x \to \pm\infty \\ x^{n-1} C \to 0 \end{cases} \tag{5.42}$$

It is easily seen from (5.41) that

$$\frac{dM_0}{dt} = 0 \tag{5.43}$$

which was expected since the transport equation (5.36) implicitly assumes that we are dealing with conservative substances. Equation (5.41) can then be expressed in a normalised form

$$\frac{dN_n}{dt} = Kn(n-1) N_{n-2} \tag{5.44}$$

By substituting the values of n=1 and n=2 in (5.44), the first and second moments of the concentration distribution are readily found to be

$$\frac{dN_1}{dt} = 0 \qquad \Rightarrow \qquad N_1 = \text{constant} \tag{5.45}$$

and

$$\frac{dN_2}{dt} = 2K \qquad \Rightarrow \qquad N_2 = 2Kt \tag{5.46}$$

Equations (5.45) and (5.46) indicate that the centroid of the distribution remains stationary, due to the non-existence of advection, while the variance changes at a rate equal to twice the diffusivity.

Since experimental observations of velocity fluctuations (Ozmidov, 1990) have shown that they follow a Gaussian (normal) distribution, which satisfies (5.45) and (5.46), the distribution from which the random step is obtained should ideally also be Gaussian. However, the central limit theorem states that, after a large enough number of timesteps, the exact form of the distribution P is irrelevant, provided that its first and second moments satisfy the relationships (5.45) and (5.46). Some commonly used functions are the two-step distribution, given by a pair of delta functions located at $\pm(2K\Delta t)^{1/2}$, and the top-hat distribution of extent $\pm(6K\Delta t)^{1/2}$; the latter type includes the uniform distributions used in numerical random number generators.

A study performed by Hunter *et al.* (1993) on a series of standard (uniform) random number generators reveals that, whereas it costs approximately twice as much in computer time to generate a random number from a normal distribution as it does from a uniform distribution, the number of timesteps necessary for a top-hat distribution to agree with a Gaussian within 10% is of order 10. Thus, when using a Gaussian distribution, the minimum resolvable time scale may be reduced by a factor of order 10.

a.2) The choice of a random number generator

Not all random number generators (RNG) are suitable for use in random walk models. Since the generation of random numbers may occupy a significant fraction of the overall simulation time in random walk modelling, it is important to find an efficient algorithm: a satisfactory generator should, first, return a sufficiently random series of numbers and, second, be computationally efficient.

Because random number generators are based on deterministic algorithms, they never generate series of truly random numbers. Moreover, since they involve numbers with a finite quantity of digits, a pattern of output values is repeated exactly after a finite cycle (generator period). Additionally, patterns may also exist over smaller scales. A good RNG should simultaneously maximise the cycle length and minimise the probability of repetitive patterns.

The general basis for standard RNG is a linear recurrence relationship in the form

$$I_{j+1} = a_{RNG}\, I_j + c_{RNG} \,(\mathrm{mod}\, M_{RNG}) \qquad (5.47)$$

where the parameters a_{RNG}, c_{RNG} and M_{RNG} are integers carefully chosen to maximise the randomness of the generated series, and to minimise the computation time. The use of RNG based on (5.47), or linear congruential generators, requires the previous input of a "seed" I_0 to initialise the random series. The resulting random sequence usually consists of numbers I_i / M_{RNG} , where I_i is a positive integer smaller than or equal to M_{RNG}.

This procedure followed to generate random numbers, however, sets an upper limit to the number of values that can be obtained before the sequence begins to repeat itself. Once I_i has taken all the integer values between 0 and M_{RNG}, the following random number, associated to $I_{M^{RNG+1}}$, will coincide with some previous value (corresponding to I_k), and all the subsequent 'random' numbers returned by the algorithm will exactly reproduce the 'random' series for I_j (j > k). Thus, the maximum period T_{RNG} of the recurrence series is equal to the module M_{RNG} in equation (5.47), although it is likely that repetition should occur before. Knuth (1981) has proven that T_{RNG} will be maximum, i.e., equal to M_{RNG}, if, and only if, a) c_{RNG} is relatively prime to M_{RNG}; b) a_{RNG}-1=b is a multiple of p, for every prime p dividing M_{RNG}; and c) b is a multiple of 4, if M_{RNG} is a multiple of 4.

Four different standard RNG (Press et al, 1992) have been tested in order to select the best for diffusion modelling. All of them are based on equation (5.47), with a value of c_{RNG} equal to zero; in this situation, the maximum possible period is $T_{RNG}=M_{RNG}-1$ (Knuth, 1981). The parameter M_{RNG} is generally chosen to be the greatest prime number that is compatible with the variable type defined in the generating algorithm. All the tested RNG return a series of strictly positive and smaller than 1 values, and none of them accept the 'seed' $I_o=0$, since then the returned value would always be zero. These generators are briefly described below, and are identified as RAN0, RAN1, RAN2 and RAN3, in order to preserve the names given in Press *et al.* (1992).

RAN0:

This generator uses the expression (5.47) with the values $a_{RNG} = 7^5 = 16807$, and $M_{RNG} = 2^{31}-1 = 2147483647$ ("Minimal Standard" generator, proposed by Park and Miller -1988-), with an approximate period $T_{RNG} = 2.15\times10^9$.

It is a fairly simple generator, which presents some problems that affect the randomness of the generated series. As an example, low-order correlations exist: if a returned random number is very small, the following value will be smaller than expected for a truly random series.

RAN1:

RAN1 is also based on a "Minimal Standard", but it shuffles the output to remove low-order serial correlations. The random deviate derived from the 'seed' I_j is not output on the jth call, but on a randomised later call. The shuffling algorithm can be seen in figure 5.5. The period T_{RNG} is approximately 10^9.

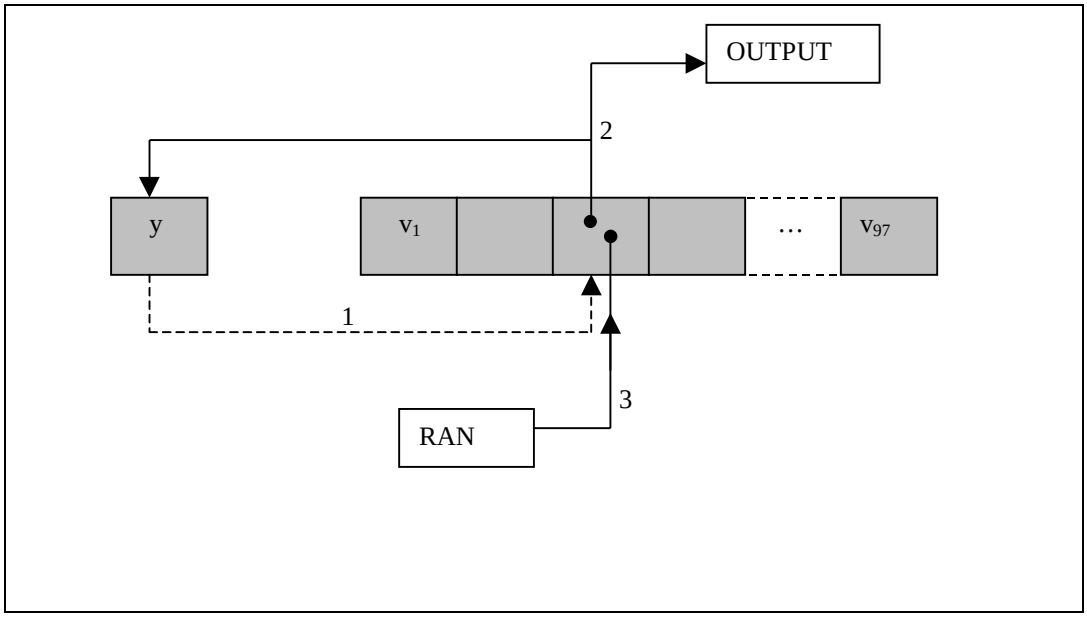

Figure 5.5.: Shuffling procedure used in RAN1 to eliminate sequential correlations. The random number in *y* is used to choose a random element in the array **v** (1); this element becomes the output random number, and is also the next *y* (2). The 'empty' space in **v** is refilled from the RNG routine (3). Figure adapted from Press *et al.*(1992).

RAN2:

RAN2 is a combination of two different sequences with different periods. On the one hand, this leads to a new sequence whose period is the least common multiple of the two individual periods and, on the other, it breaks up serial correlations to a great extent. Nevertheless, this generator includes an additional shuffling procedure. Its period is very long, greater than 2×10^{18}.

RAN3:

The fourth RNG tested is not based on a linear congruential method but rather on a subtractive method (Knuth, 1981).

There are several ways of testing the suitability of a random number generator. Hunter *et al.* (1993) suggested comparing the rate of growth of a patch of modelled particles with the theoretical relationship -equation (5.46)-. This test has been performed on the four random number generators described above, with a diffusivity K=1/24, a timestep Δt=1, 10^3 particles and 10^4 timesteps. According to equation (5.46) the variance of the required distribution should be 1/12, which is provided by a top-hat distribution between ± 0.5. For N particles, the patch variance is defined as

$$\mathrm{var} = \frac{1}{N}\sum_{i=1}^{N} x_i^{\,2} \tag{5.48}$$

where x_i is the coordinate of particle i. The ratio of the standard deviation of the patch variance to the mean patch variance can be found from the properties of the χ^2 distribution to be $(2/n)^{1/2}$.

The dashed lines in figure (5.6) represent one standard deviation from the theoretical variance, i.e.,

$$\frac{t_n}{12}\left(1 \pm \left(\frac{2}{n}\right)^2\right) \tag{5.49}$$

where t_n is the timestep number, and n the degrees of freedom. It can be seen that there is a significant difference between the four tested RNG, and that not all of them are acceptable in a random walk modelling of turbulent diffusive motion.

Vattulainen and Ala-Nissila (1995) have proposed an alternative method for which they use the chi-square test with three degrees of freedom. The two-dimensional random walker defined by these authors starts at the position (0,0) and jumps a unit distance each timestep, in any of four possible directions with equal probability 1/4; the actual direction is determined by the random numbers. After N steps, the random walker should have jumped in each direction an equal (approximately) number of times. The number of occurrences in each direction O_i (i=1, 2, 3, 4) is computed and compared with the expected value E_i=N/4 using the χ^2 test:

$$\chi^2 = \sum_{i=1}^{4} \frac{(O_i - E_i)^2}{E_i} \tag{5.50}$$

After studying the upper-tail of the chi-square distribution, the authors decided that a RNG fails the test if the χ^2 value exceeds 7.815 in at least two out of three independent runs. The following table shows the value of χ^2 obtained for each one of the tested RNG in a series of nine independent runs. It can easily be seen that, although all the generators pass this test, RAN2 seems to obtain the best values of χ^2.

I_o	RAN0	RAN1	RAN2	RAN3
-1	4.195	2.185	0.862	3.208
-100	0.911	6.988	0.746	2.791
-115	2.315	3.994	3.438	0.386
-300	2.071	3.599	5.414	2.795
-3000	4.164	5.297	6.167	1.644
-73921	2.439	7.309	0.908	1.506
-100324	2.374	4.345	0.915	1.077
-115623	3.291	3.969	0.454	5.682
-300001	9.826	4.355	1.362	4.716

Table 5.1: Values of χ^2 in nine independent runs, for each one of the tested RNG. The first column indicates the RNG initialising "seed"- I_0 - used in each run.

Finally, the CPU-time consumed by each RNG in yielding a single random number has also been computed, using a DIGITAL ALPHA 7630AXP computer, and the obtained results are shown in table 5.2:

	Time (μsec)
RAN0	0.37
RAN1	0.50
RAN2	0.96
RAN3	0.13

Table 5.2: Time required to generate a random number, for each one of the tested RNG.

Since it is preferred to favour randomness rather than computational speed (although it is recognised that speed is a relevant feature of any numerical model, the constant development of faster computers gives this variable a relative importance) a greater weight was given to the results of the tests proposed by Hunter *et al.* (1993) and Vattulainen and Ala-Nissila (1995) than to tests based on "number-generation time" when selecting the RNG for the transport model. Therefore, RAN2 was chosen, since it appears to perform the best in both Vattulainen and Ala-Nissila's (1995) and Hunter *et al.*'s (1993) tests. Figure 5.7 shows the distribution of a series of 50,000 random numbers generated by RAN2, and it can be seen that the number distribution appears to be quite uniform.

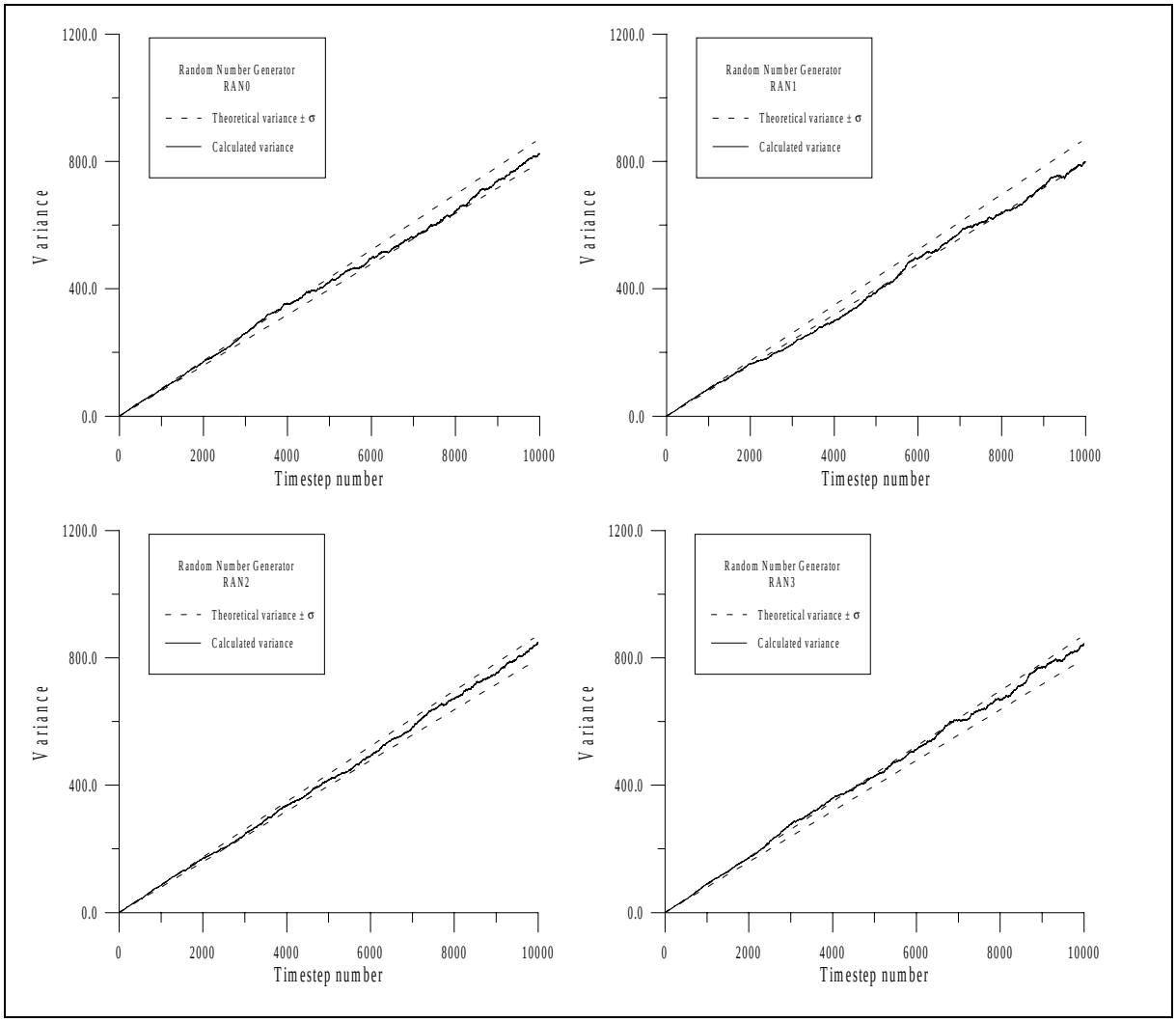

Figure 5.6: Results of the tests of RNG. The continuous curve represents the variance of each simulation; the dashed lines indicate one standard deviation on either side of the theoretical variance.

a.3) Mixing coefficients

To compute the transport due to turbulent and molecular diffusion it is first necessary to define the mixing coefficients that describe diffusive processes. Molecular diffusion is assumed isotropic, and is described by a single coefficient, but mixing induced by the turbulent nature of the flow is generally anisotropic, and at least three parameters must be specified in order to describe it.

a.3.1) *Molecular diffusion coefficients*

The molecular diffusivity is a variable that does not depend at all on the flow conditions, but is a function only of the solute and the solution. Although its value is much smaller than turbulent diffusion coefficients, it has been included in the model for the sake of completeness. The typical values found are of the order of 10^{-9} m^2/s.

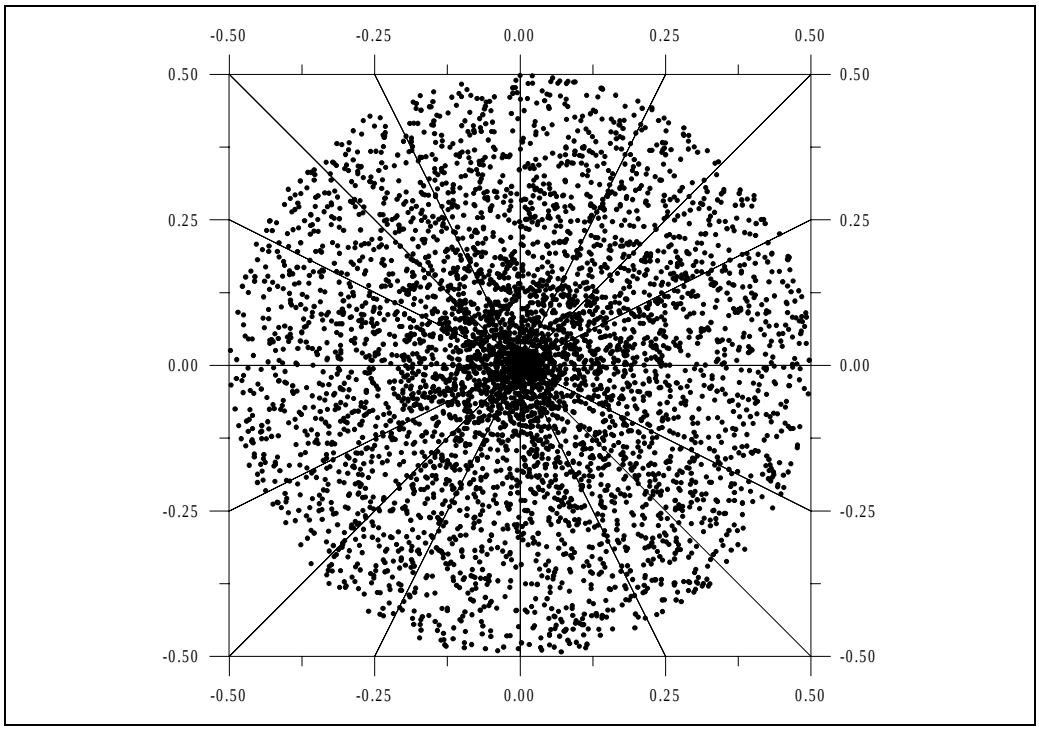

Figure 5.7: Distribution of a set of 50,000 random numbers generated by RAN2.

a.3.2) Turbulent diffusion coefficients

As was said in §2.3.1.2, it is possible to define turbulent diffusion coefficients in the directions parallel and normal to the mean flow direction, in addition to a vertical diffusivity. Although it is generally assumed that the horizontal coefficients are at least an order of magnitude larger than the vertical coefficients, Svendsen and Putrevu (1994) have shown that inside the surf zone K_H and K_V are of the same order, but that horizontal mixing is larger owing to an additional mixing mechanism related to the vertical variations of the cross-shore and longshore currents, and their interaction. Nevertheless, the widespread "horizontal-vertical" decomposition is adopted here.

In the horizontal plane, the expressions used are of the type proposed by Elder (1959) and Holly (1985)

$$\begin{aligned} K_L &= c_L u_* h \\ K_T &= c_T u_* h \end{aligned} \qquad (5.51)$$

where c_L and c_T are experimental coefficients. Elder (1959) proposed c_L = 5.91 and c_T = 0.23; although these expressions are derived for two-dimensional flows, they are commonly used in 3D situations to describe the horizontal dispersion. It has been shown, however, that the longitudinal dispersion coefficient given by Elder (1959) significantly underestimates natural dispersion, because it does not consider the transverse variation of the velocity profile (Seo *et al.*, 1998).

In the surf zone, equations (5.51) cannot be applied, since turbulence levels are enhanced by the transfer of wave energy to turbulent eddies due to wave breaking, and this phenomenon is not included in Elder's expressions. Moreover, some authors (e.g., Rodriguez, 1997) consider horizontal turbulent mixing to be isotropic in this region, so a single diffusion coefficient can be used. Several expressions

are found in the literature to describe surf zone turbulent mixing, amongst them those due to Thornton (1970)

$$K_H = \frac{H_w^2}{8\pi^2}\frac{gT_w}{h}\cos^2\alpha_w \tag{5.52}$$

and Battjes (1983)

$$K_H = c_H\left(\frac{D_B}{\rho}\right)^{1/3} h \tag{5.53}$$

where T_w stands for the wave period, H_w is the wave height, α_w is the wave incidence angle, D_B is the rate of wave energy dissipation (see §2.3.1.4), ρ is the water density, and c_H is a constant.

The vertical turbulent diffusion coefficient outside the surf zone is computed using an expression similar to equation (5.51), but with a different constant c_V, ranging between 0.16 and 0.23. Inside the surf zone, the increase in turbulence levels is accounted for by using an expression proposed by Nadaoka and Hirose (1986):

$$K_V = 0.18\left(\frac{H_w}{h}\right)^{7/3} L_w^{1/3} h^{2/3}(gh)^{1/2} i^{1/3} \tag{5.54}$$

where L_w is the wavelength, and i is the bottom slope.

These coefficients must be substituted into equation (5.35) in order to find the diffusion velocity component.

5.2.1.3 Buoyancy term

The vertical velocity component induced by the differences in density between the discharged effluent and the receiving waterbody must be accounted for in 3D transport models, specially when these are to be used in cases such as marine outfall discharges. The buoyancy-induced shear at the plume interface leads to the entrainment of ambient water within the body of the plume, reducing the ratio between the plume and the environment density, which is a key factor in determining whether the plume will reach the water surface or will remain trapped at a certain depth.

The treatment of plume buoyancy is not straightforward in Lagrangian models, because these are not well suited to reproduce volume forces; in this particular case, Lagrangian-based models are unable to simulate correctly the entrainment phenomenon (Anfossi *et al.*, 1993). The usual approach found in the literature consists in the inclusion of an additional vertical velocity in the random walk equation (Zannetti and Al-Madani, 1983; Anfossi *et al.*, 1993; Luhar and Britter, 1992), although some authors attempt to describe the buoyancy of each particle as a function of its height above the source and its radial distance from the plume centreline (Cogan, 1985).

The approach adopted here is the one presented by Anfossi *et al.* (1993). They include a vertical buoyant velocity component in the random walk equation, as a function of the discharge density, the local environmental density gradient, and the time elapsed since the particle was released; the plume self-induced turbulence is accounted for by providing some randomness to the buoyancy velocity, following a given probability distribution. The velocity they define is

$$w_B = \frac{H_p(u,s,t+\Delta t) - H_p(u,s,t)}{\Delta t} \tag{5.55}$$

where H_p describes the rise of the plume centreline with time. The expression for the plume rise proposed by Anfossi (1985) is also adopted,

$$H_p(u,s,t) = 2.6\left(\frac{\bar{F}\,t^2}{\bar{u}}\right)^{1/3}\left(t^2 s + 4.3\right)^{-1/3} \tag{5.56}$$

which is a generalisation of the classical Briggs's (1975) formula. In this equation, $\bar{u}$ is the mean current velocity, and $\bar{F}$ and s are defined as

$$\bar{F} = g w_0 r^2 \frac{\rho_0 - \rho_a}{\rho_0} \quad ; \quad s = \frac{g}{\rho_a}\frac{\partial \rho_a}{\partial z} \tag{5.57}$$

with w_0 the discharge velocity, r the outfall radius, and ρ_0 and ρ_a the initial discharge density and the ambient density, respectively. Based on experimental observations (Anfossi *et al.*, 1993), a normally distributed $\bar{F}_i$, with mean value $\bar{F}$ and standard deviation equal to $\bar{F}$ /3, is assigned to each particle in order to represent the plume self-induced turbulence, which decreases progressively, affecting only the vertical dispersion. Figure 5.8 plots equation (5.56) together with the modelled plume centreline corresponding to two different discharge velocities.

For stagnant environments, equation (5.56) can not be used, and a simple expression for buoyancy-induced velocities based on basic fluid dynamics is implemented:

$$w_B = \frac{1}{2} g \frac{\rho_0 - \rho_a}{\rho_0} \Delta t \tag{5.58}$$

5.2.1.4 Particle resuspension term

In accordance with the Lagrangian formulation, the modelling of sediment suspension is done by lifting particles up from the bed, after specifying a vertical suspension velocity w_R. This velocity is estimated by taking advantage of the pickup function, which is a sediment mass flux, dimensionally equivalent to a concentration times a velocity:

$$E_r \propto C w_R \tag{5.59}$$

In order to reduce the memory and computational time required, sediment suspension is computed only in those cells for which the condition of motion initiation holds, i.e.,

$$\tau_b(i,j) > \tau_{b,cr}(i,j) \tag{5.60}$$

The total bed shear stress is computed using equation (3.59); the current-related and wave-related shear stresses at the bottom are calculated using equations (3.50) and (3.53), respectively. The corresponding friction factors are given by equations (3.52) and (3.56), and the Chezy coefficient given in (3.51) is also used. The critical shear stress is found by assuming a value of θ_{cr} = 0.05 (Nielsen, 1992), and solving equation (3.71).

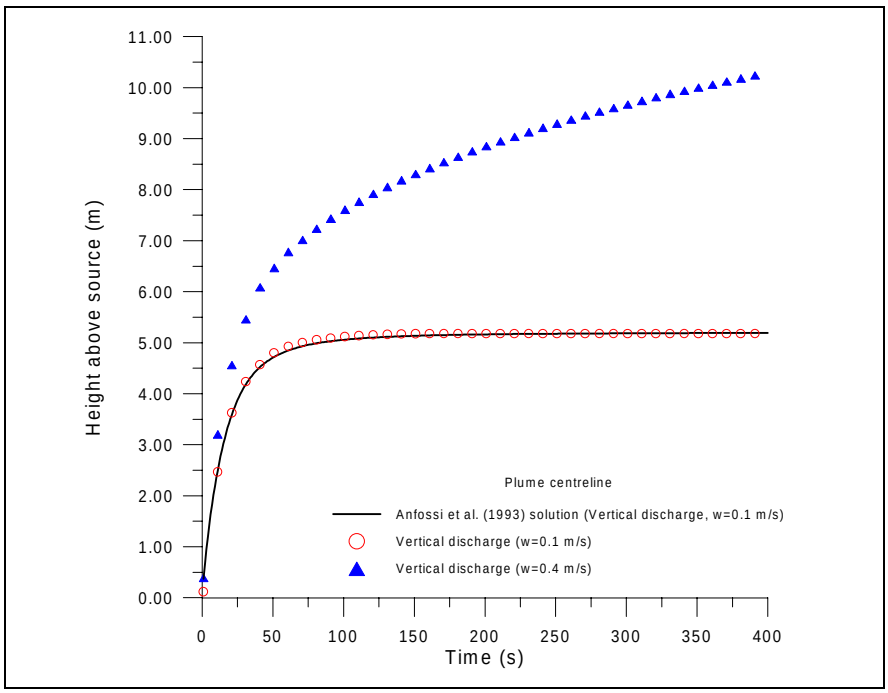

Figure 5.8: Comparison of modelled and theoretical -equation ((5.56))- plume centreline rise, for two different discharge velocities.

For each one of the cells where sediment motion will occur (active cells), the bed-load concentration is then calculated using Van Rijn's (1984a) expression –equation (3.79)-, and the value of the pickup function –equation (3.114)- at each cell is also computed, from where the suspension velocity is obtained:

$$w_R = a_1 \frac{E_r}{C_B} \tag{5.61}$$

where a_1 is a proportionality coefficient that must be obtained by calibration with experimental data. The vertical velocity so calculated is entered into the particle equation of motion as an additional "advective velocity" for particles belonging to the bed-load layer (i.e., $z = z_B$). Further upward, this velocity is zero, and the particle transport is due only to ambient currents and turbulence, and to the effects of the particle settling velocity. A suspension maintaining mechanism for the sediments should also be considered, but is not yet implemented in the numerical model.

It remains to determine the number of Lagrangian elements that are entrained into the flow above each cell. For this purpose, a sediment mass is calculated in each computational cell as

$$M_S = C_B \Delta X \Delta Y \delta_B \tag{5.62}$$

where δ_B is a height from the bed in which C_B is assumed to be constant. To be coherent with the model chosen to compute the bed-load concentration, δ_B is taken to be the particle saltation height

given by equation (3.82). Since it is unpractical to use the real number of sediment particles, a predefined minimum number of particles $N_{S,min}$ is assigned to the cells with the smallest sediment mass, $M_{S,min}$, while the particle number for the remaining grid cells is calculated as a function of the mass ratio, $N = N_{S,min}M_S/M_{S,min}$.

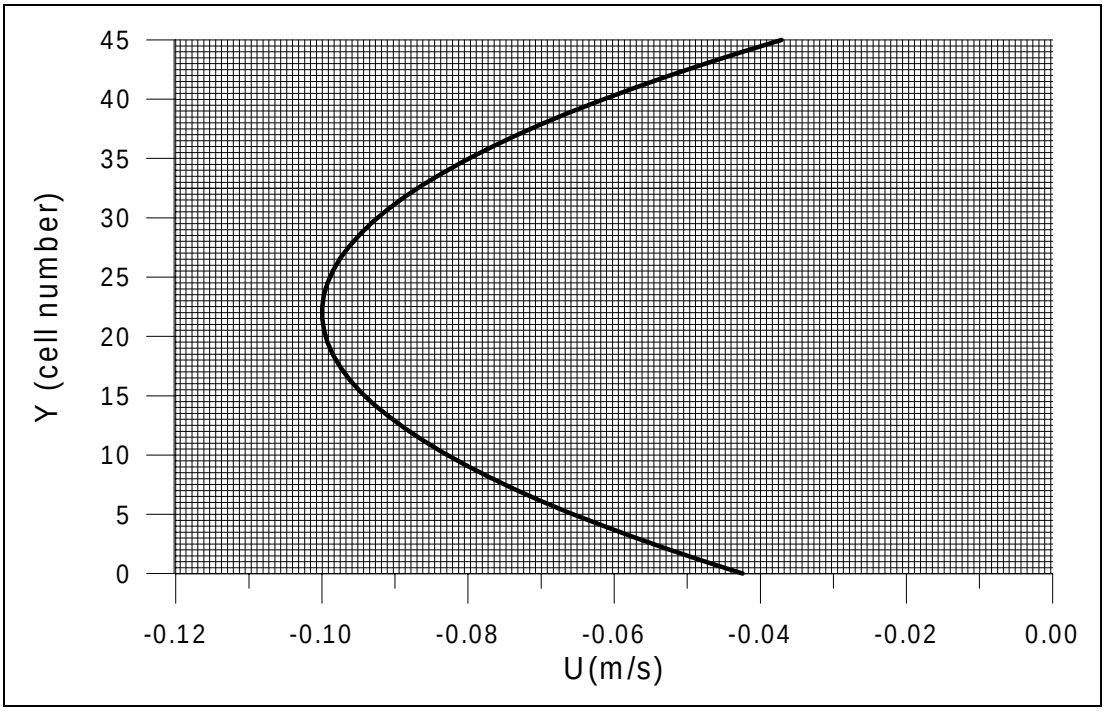

Figure 5.9: Flow profile used for the resuspension test.

A simple test was carried out to estimate a reasonable value for $N_{S,min}$, considering a rectangular basin, with a constant depth of 17m and a superimposed grid of 75x45 cells, each one of size ΔXxΔY. The water current within this domain was taken to be parabolic in the *y*-direction, and uniform in the *x*- and *z*- directions (figure 5.9), so that particle resuspension occurred only in the central cells along the *x*-axis. The diffusion coefficients were taken to be $c_L = 0.00593$, $c_T = 0.021$, and $c_V = 0.00001$; a very small value for c_V was chosen to avoid the masking of the relevant resuspension and settling processes by vertical turbulent diffusion. The sediment was assumed homogeneously distributed over the whole bed, with a density of 2650 kg/m^3, and d_{50} = 0.001m, whereas the water density was assumed uniform and equal to 1026.5 kg/m^3. Particle transport was simulated during 120s, with a 1s timestep. The morphological variations of the flat bottom were calculated as the difference between the resuspended and the sedimented particles within each cell, using different values for $N_{S,min}$, and an error was defined by comparing the results for each run with those from the run with the largest $N_{S,min}$ (in this case, 60).

A plot of this error, for ΔX = ΔY = 2m is given in figure 5.10, from where it is seen that, beyond $N_{S,min}$=20, the increase in accuracy does not appear to be justifiable in terms of CPU-time and number of particles –i.e., computational memory- involved in the model run. A very similar behaviour is observed (figure 5.11) using larger cell sizes (ΔX = ΔY = 10, 15, 20, 30, 50 and 99 m), and the value $N_{S,min}$=20 is therefore adopted in the model.

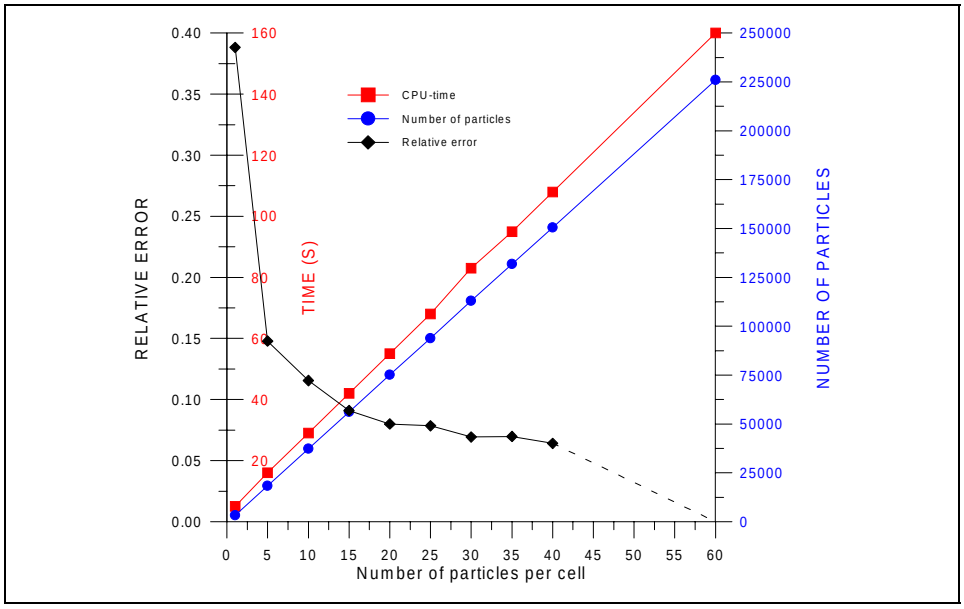

Figure 5.10: CPU-time usage, number of particles in the simulation, and error as a function of $N_{S,min}$.

A second test has been done, with the same configuration, but changing the size of the computational cells while holding $N_{S,min}$ fixed and equal to 20. The differences in the erosion profiles are significant only near the ends of the domain, because the particles resuspended from the smaller cells are dragged away to other cells by the current before they can settle, whereas those from large cells have time to settle in the cell they originated from. Away from the boundaries, the profiles are similar independently of the size of the computational cells.

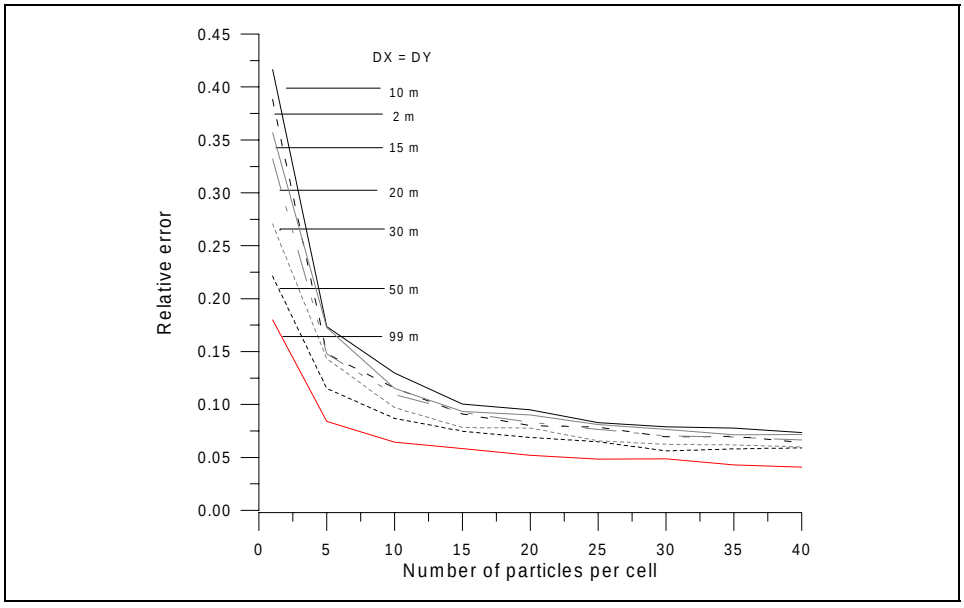

Figure 5.11: Relative error as a function of $N_{S,min}$, for different grid cell sizes.

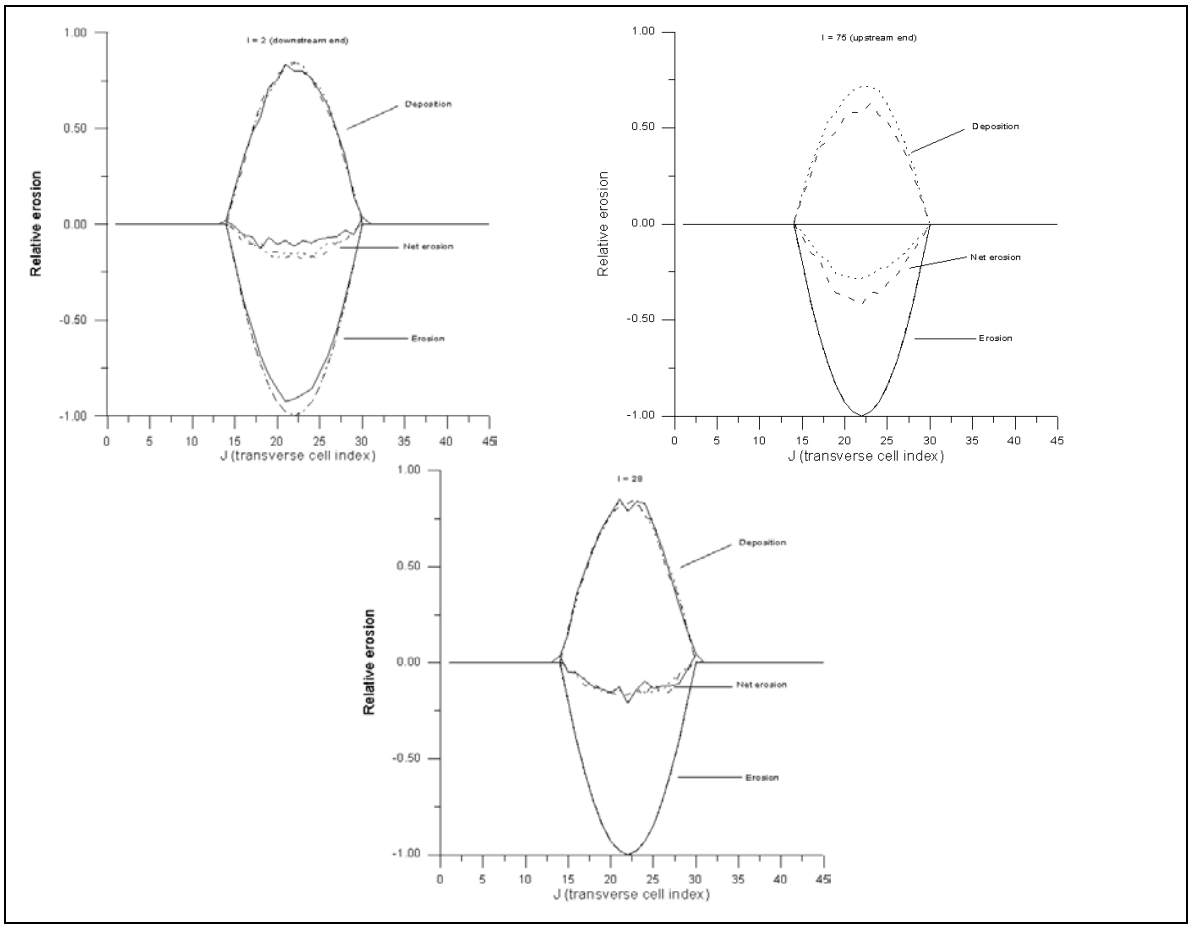

Figure 5.12: Erosion profiles at different locations within the domain, for different computational cell sizes. Solid line, ΔX = 2m; dashed line, ΔX = 15m; dotted line, ΔX = 30m.

The method developed here for the modelling of sediment resuspension appears to be intuitive and conceptually simple, allowing for its easy implementation within the Lagrangian framework. On the other hand, the number of particles involved in the computations may be extremely large, depending on the size of the computational grid and the magnitude of the currents, leading to CPU-time and computer memory problems.

5.2.1.5 Particle settling term

Due to the existence of heavy particles in most substances discharged into coastal waters, it is important for transport models to include a settling module in order to have an estimate of the transport and final distribution of particles on the waterbody bed. In LIMMIX, particles are treated as spheres, and semi-empirical equations are applied to each one to obtain the settling velocity used in equation (5.15). In particular, the expression obtained by Gibbs *et al.* (1971) is used (equation 3.121), together with the correction introduced by Richardson and Zaki (1954) –equation (3.125)- to account for the effects of sediment concentration.

5.2.1.6 Microbiological inactivation

The modelling of the microorganism decay process is done by using equation (3.2) or (3.4) in each cell of a Eulerian three-dimensional mesh superimposed on the hydrodynamic computational grid. It is assumed that the physical properties of the water (and, thus, the decay coefficient) present a very slight dependence, or none at all, on the horizontal coordinates (x,y), so that a low resolution is sufficient in the horizontal plane. In the vertical direction there is a significant change in the decay rate due to the attenuation of the solar radiation with depth, and a higher resolution is therefore required; in addition, since light extinction takes place mainly in the upper layer of the water body, the horizontal layers of the "decay grid" are chosen to be thin near the surface, and thicker further down the water column (figure 5.13a).

Under this assumption, a statistical approach is used in which, out of the total number of particles $N(i, j, k)$ contained in a "decay cell" (i, j, k) at a time t_n, only $N\_(i, j, k)$ are taken into account at time t_{n+1}. The value of $N\_(i, j, k)$ is given by

$$N\_(i,j,k) = N(i,j,k)e^{-K_d(i,j,k)\Delta t} \tag{5.63}$$

or

$$\log N\_(i,j,k) = \frac{K_d'(i,j,k)}{K_d'(i,j,k) + t_n} \log N_o(i,j,k) \tag{5.64}$$

depending on whether the chosen rate of decay is linear or hyperbolic, and where K_d (i,j,k) and K'_d (i,j,k) are the respective decay coefficients, assigned to the (i,j,k) cell.

a) Calculation of the decay rate

The decay rates appearing in equations (5.63) or (5.64) may be introduced in the computations in three different manners:

a) K_d (or K_d') is an input parameter, constant both in time and space.

b) K_d (or K_d') is a constant throughout the computational domain, but is derived from experimental data as a function of time of the year and type of bacteria involved. The data (T_{90} = ln10/K_d or K_d') used in this case are from Bravo (1985), obtained from experiments in the Mediterranean Sea, and are shown in tables 5.3 and 5.4, in which "FS", "FC" and "TC" stand for "faecal streptococci", "faecal coliforms", and "total coliforms", respectively. A sunlight factor is introduced to account for the hourly variations of the decay rate, based on Mead and Cooper (1992) and adopted from Bahia (1997), and is given in table 5.5.

Month / Indic.	J	F	M	A	M	J	J	A	S	O	N	D
FS	15	17	33	33	50	75.5	100	25.5	33	25.5	33	14.5
FC	13	14.5	20	26	26	37.5	50	15	18	20	18	11
TC	12	12	14	25	25	29	33	14.5	17	17	17	12

Table 5.3: Values for T_{90} [min] for different types of indicators (Bravo, 1985).

Month Indic.	J	F	M	A	M	J	J	A	S	O	N	D
FS	29	28	22	46	172	178	183	4.5	40	11	40	31
FC	25	26	10	28	51	63	75	6.5	25	4.5	23	25
TC	33	33	14	38	68	85.5	106	11	31	6	28	33

Table 5.4: Values of K_d' [min] for the hyperbolic inactivation model, for different types of indicators, calculated from field data (Bravo, 1985).

HOUR	SUNLIGHT FACTOR
1	2.31
2	2.31
3	2.31
4	2.31
5	2.31
6	1.15
7	0.52
8	0.38
9	0.26
10	0.25
11	0.23
12	0.25
13	0.26
14	0.29
15	0.33
16	0.38
17	0.40
18	0.49
19	0.58
20	0.70
21	0.92
22	1.15
23	1.62
24	2.31

Table 5.5: Sunlight factor applied to constant decay rates to account for the hourly variation of the solar radiation effects, based on Mead and Cooper (1992).

c) K_d (or K_d') is a function of both time and space. In this case, the expressions for the decay rate due to Mancini (1979) -equations (3.5) and (3.6)- and Juanes (1995) -equations (3.5) and (3.8)- may be used, together with equations (3.10) to (3.17) for the calculation of solar radiation intensity at any depth.

In the former two cases the decay rate at each "decay cell" is equal to the constant K_d (or K'_d). In the latter situation $K_d(i,j,k)$ is taken as the mean value of K_d (or K'_d) at each node of the cell (figure 5.13b), and is valid only for the first-order decay pattern as given by equation (5.63):

$$K_d(i,j,k) = \frac{1}{8}\sum_{m=1}^{8} K_d^{(m)}(x_m, y_m, z_m) \tag{5.65}$$

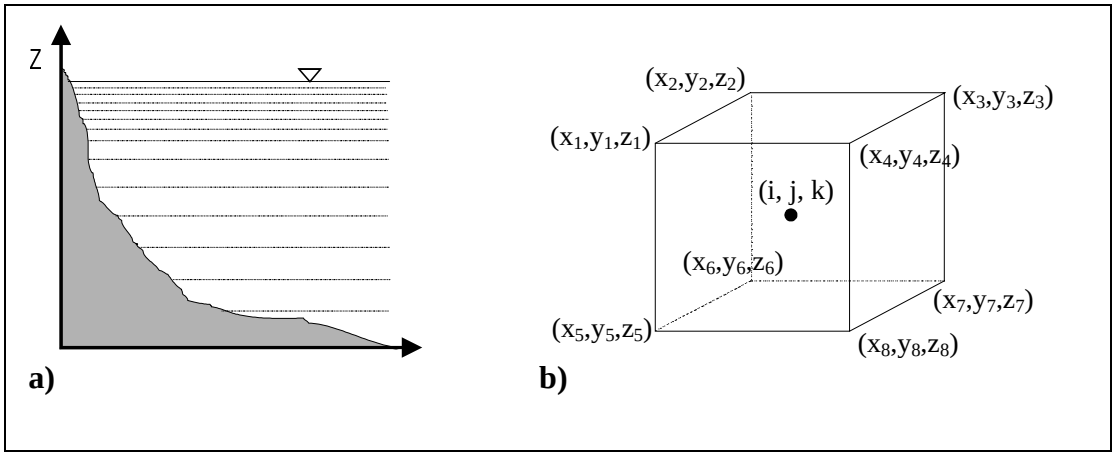

Figure 5.13: a) Vertical cell separation for the decay mesh. **b)** Decay cell nodes used to compute the decay rate for cell (*i*,*j*,*k*).

5.2.1.7 Boundary conditions

The application of boundary conditions to pollutant transport within a particle model is a straightforward and simple operation. Because of the discretisation introduced by the Lagrangian solution method, all that is required is the specification of the behaviour of any Lagrangian element when it contacts the boundary, i.e., if the particle crosses the boundary, or if it returns to the computational domain.

In a general situation, the computational domain where a particle can be located is vertically bounded by two surfaces that correspond to the free-surface and the sea bed, whereas one or more 'dry' surfaces -coast, artificial structures-, or one or more non-physical planes defined by the problem requirements -for instance, the seaward limit of the computational grid- constitute the horizontal boundaries (figure 5.14). A particle contacting any of these surfaces may behave in a number of different ways: the non-physical (open) boundaries are not real limits, so the particle will cross them, simply disappearing and counting as lost mass; these boundaries must be far away enough from the sources to allow the possible contribution from returning pollutants to be neglected. On the other hand, the free surface generally acts as an impermeable boundary, although it can be permeable to a certain degree if the Lagrangian element represents heated water. Finally, the solid boundaries can be considered anything between fully particle-absorbent and completely non-absorbent, depending on a reflectivity coefficient.

The approach followed in this Lagrangian model defines only horizontal boundaries for the non-physical vertical planes, since the remaining may be reduced in an adequate manner to vertical boundaries; amongst these, the solid boundaries –coast and sea bottom- and the free surface boundary are treated separately.

Each one of the 2DH computational grid cells is characterised by the water depth h_{ij} and a solid boundary reflectivity coefficient a_{ij}; the free surface is considered, for the moment, fully non-permeable

($a_{ij} \geq 0$), and the "open" limits of the computational grid are taken as particle sinks. Obviously, dry land cells have h_{ij} equal to zero.

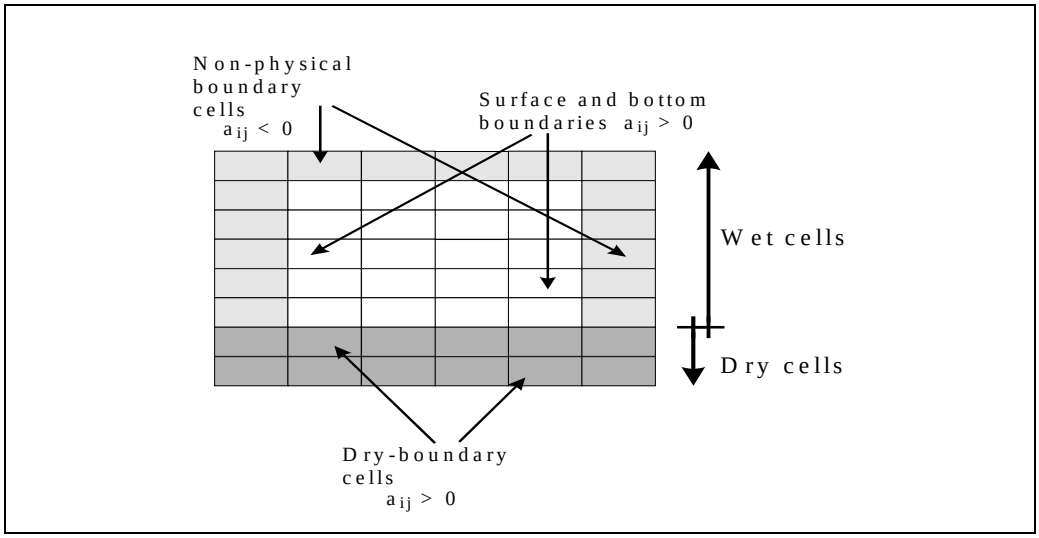

Figure 5.14: Types of boundaries considered in the transport model.

a) Reflectivity coefficient

The reflectivity coefficient used here is defined as the percentage of particles that return to the computational domain after contacting a solid boundary. The lowest real value of a_{ij}, that is, $a_{ij} = 0$, corresponds to boundaries which absorb all the particles, which may therefore accumulate on the boundary surface, and its largest value ($a_{ij} = 100$) is associated to boundaries which do not absorb any of the impinging particles. It is clear that the reflectivity coefficient depends strongly on the type of particle (on the type of substance that is modelled) and on the nature of the solid boundary (rock, sand, clay, etc).

In order for the transport model LIMMIX to run properly, the reflectivity coefficient should be defined for each cell of the computational grid. The model allows the definition of reflective coefficients smaller than zero, to account for permeable boundaries. In this case, the numerical value assigned to a_{ij} is not important, since it only states that particles are lost through the boundary, and disappear from later calculations.

b) Implementation of boundary conditions

- Non-physical boundaries:

  These boundaries are assumed completely permeable, so any particle that exits the computational grid will automatically disappear from subsequent computations. The total mass inside the domain will consequently decrease.

- Free surface:

  The free surface is actually considered only as an impermeable boundary. Any particle whose vertical coordinate z becomes negative is moved to $z = 0$, without varying its horizontal coordinates (x,y).

- Solid boundaries:
  Two cases, illustrated in figure 5.15, can be discussed herein:

  - Dry cells: With h_{ij}=0, they represent obstacles that can be either natural (coasts) or artificial (breakwaters, etc). If a particle ends up in a dry cell with an index equal to one of the particle's origin cell, it is given a new horizontal position which is the intersection point (x_{exit},y_{exit}) of the dry cell's edge and the particle's path, defined by (x_o,y_o) and (x_1,y_1), as can be seen in (figure 5.15a).

    If both cell indexes differ, LIMMIX determines in the first place the obstacle 'orientation'. If it runs in the x direction, a y_{tall} is defined -(2) in figure 5.15b which will also be the particle's y-coordinate, and the x-coordinate is calculated from the intersection of the particle path and $y = y_{tall}$. If the obstacle runs in the y direction -(1) in figure 5.15b-, the intersection of the particle path and $x = x_{tall}$ yields the particle's final horizontal position. The vertical coordinate is taken to $z = 0$ in both cases.

    In case the obstacle is composed of a single cell, that is, it has no orientation, the particle remains in its initial position.

  - Wet cells (bottom boundary): Where h_{ij} is positive. Any particle that reaches a vertical position z greater than the water depth will take a new position $z = h_{ij}$, without changing coordinates (x,y).

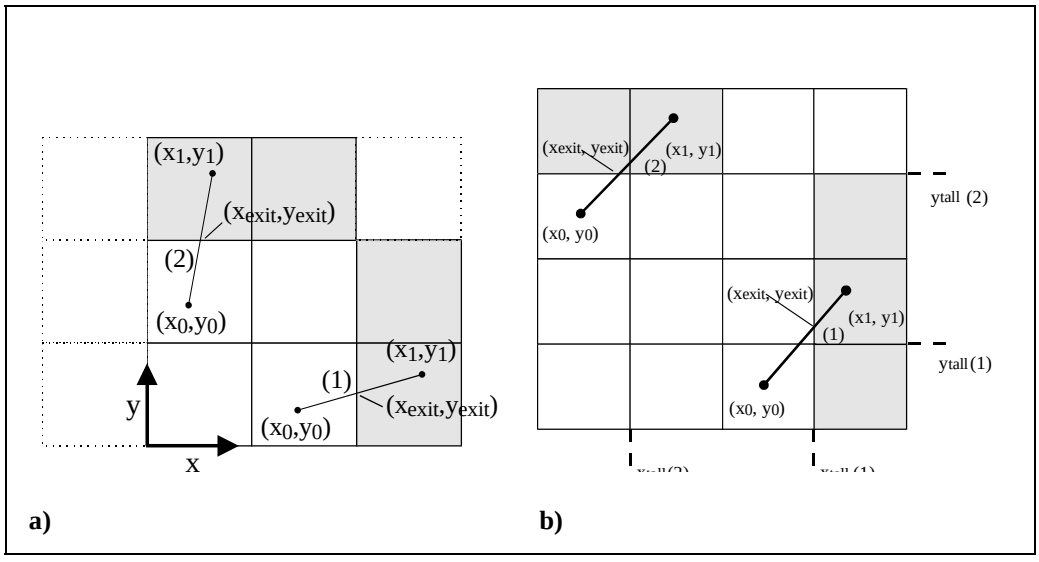

Figure 5.15: Implementation of boundary conditions in the transport model.

From all the particles existing at a given timestep t_n, those affected by the physical boundary conditions are tagged with a positive boundary index -IPOS(4)-, whereas for the rest IPOS(4) = 0. At the following timestep t_{n+1} only a certain number of particles with IPOS(4) > 0 will be moved, depending on the reflectivity coefficient. The number of particles in a cell (i,j) with IPOS(4) > 0 at a time t_n that will keep moving at time t_{n+1} is given by

$$\text{No. of particles to be moved} = \underset{(NCOMP)}{\text{No. of particles in cell}} \cdot \text{reflectivity coefficient} \qquad (5.66)$$

Only the NCOMP first particles in cell (*i*,*j*) will be moved at the following timestep; the remaining particles will 'freeze' in their last position, until the end of the simulation is reached.

5.2.1.8 Computational timestep

The computational timestep that is used in the model runs must be chosen carefully, since it is an important parameter in determining the accuracy of the solution. In general, the selected timestep should have an upper limit related to the resolution of the hydrodynamic computational grid, so as to force the Lagrangian particles to follow the current lines defined by the velocity field as smoothly as possible. Certain particular situations, however, may have the effect of lowering the value of this limit; examples of these cases are

a) Near physical boundaries (such as the coast or man-made structures), where abrupt changes in the direction of the current vectors may appear,

b) In regions where the velocity field naturally presents strongly curved flowlines.

c) In bodies in which the watercolumn shows some degree of stratification.

In the latter case, the use of a large timestep will lead to particle overshooting at the picnocline -even when the numerical correction to buoyancy due to water entrainment is computed-; occasionally, the interface will be “ignored” by the particles, and these will behave as in a uniform watercolumn. In the particular case of a river discharge, where the freshwater velocity at the river mouth can be much larger than the mean longshore velocity, a large timestep may result in particles accumulating at some distance from the river mouth, since the Lagrangian elements reach this region at a larger rate than they leave it, dragged by coastal currents; this leads to unreal concentration maxima away from the discharge point.

Mechanism	Name	$\Delta t <$	Observations
Horizontal advection	$(\Delta t)_{ah}$	$\frac{\min(\Delta x, \Delta y)}{\max(u_i)}$	The particle displacement in a Δt due to advection is smaller than the smallest dimension of the computational cells.
Horizontal diffusion	$(\Delta t)_{adh}$	$\frac{4[\min(\Delta x, \Delta y)]^2}{6\max(D_L, D_T)}$	The particle displacement in a Δt due to turbulent diffusion is smaller than the smallest dimension of the computational cells.
Vertical advection	$(\Delta t)_{av}$	$\frac{\Delta z}{\max(w_i)}$	The vertical displacement of a particle in a Δt due to advection is smaller than the vertical grid resolution, for 3D HD fields.
Vertical diffusion	$(\Delta t)_{dv}$	$\frac{4\Delta z^2}{6D_0}$	The vertical displacement of a particle in a Δt due to turbulent diffusion is smaller than the vertical grid resolution, for 3D HD fields.
Molecular diffusion	$(\Delta t)_{dm}$	$\frac{4[\min(\Delta x, \Delta y)]^2}{6D_0}$	It is not relevant in practice, although it is conceptually valid.

Table 5.6: Physical criteria imposed on the computational timestep used in model runs.

Mechanism	Name	$\Delta t <$	Observations
Buoyancy	$(\Delta t)_b$	$\left(\frac{6D_V}{g^2 \left(\frac{\Delta\rho}{\rho} \right)_0^2} \right)^{1/3}$	Imposed so that displacement due to density deficiency is not greater than spreading due to vertical diffusion. It avoids the appearance of vertical pulses when the discharge is continuous.
Settling	$(\Delta t)_s$	$\frac{6D_V}{4w_S^2}$	Imposed so that displacement due to settling is not greater than spreading due to vertical diffusion.
Horizontal advection	$(\Delta t)_h$	$\frac{6\max(D_L, D_Y)}{4[\max(u_i)]^2}$	
Vertical advection	$(\Delta t)_v$	$\frac{6D_V}{4[\max(w)]^2}$	Only for 3D or Q3D velocity fields.

Table 5.7: Aesthetic criteria imposed on the computational timestep used for model runs.

Several criteria have been established in order to obtain an accurate modelling of the transport process, based on both physical and aesthetic rules. The former govern the behaviour of the plume (i.e., they guarantee that the pollutant will remain trapped in stratified waterbodies, that the plume will follow the current lines, etc), and are therefore the most important criteria (table 5.6); the latter are introduced simply to ensure the continuity of the resulting concentration field, and to avoid the appearance of pollutant pulses as a consequence of treating a continuous source with a discrete timestep. Aesthetic criteria are given in table 5.7.

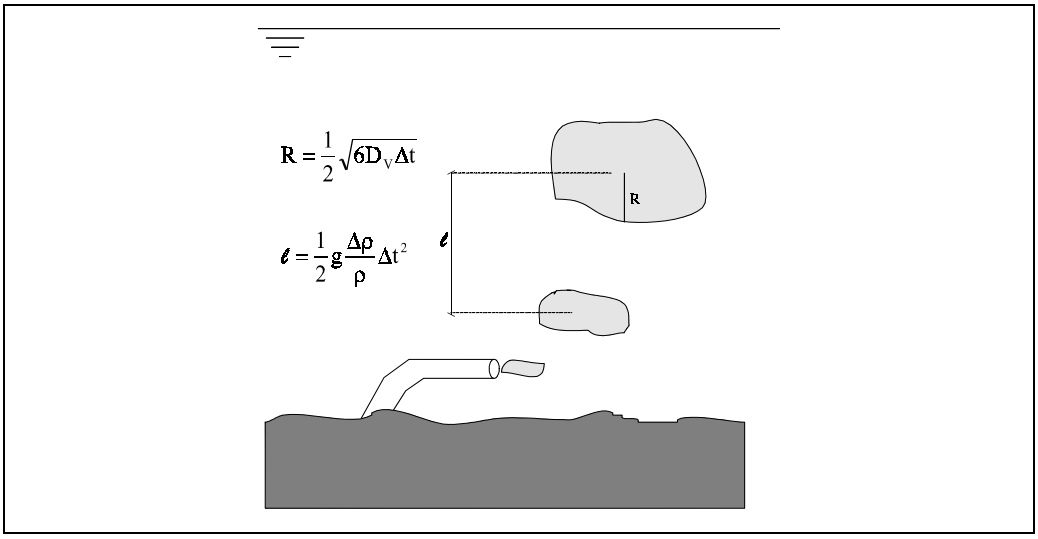

Figure 5.16: Scheme followed to derive the limit to computational timesteps imposed by buoyant displacement.

The equations in table 5.7 are found by setting the conditions under which the vertical displacement of the centre of a pollutant pulse during a timestep is not greater than the spreading of the pulse because of turbulent diffusion. This is illustrated in figure 5.16, where the procedure followed to derive the buoyancy timestep criteria is described.

5.2.2 Mapping particles onto grid nodes

From a practical point of view, we are generally more interested in obtaining concentration values at the nodes of a defined grid than mere discrete particle positions. Therefore, a means of transferring the information contained in the particle distribution to a set of selected spatial points is needed. It should be recalled that, because of the stochastic nature of random walk particle methods, the concentration estimates made from the particle distribution include a "noisy" error component, which in some cases may be propagated forward in time. This error can be reduced by increasing the number of particles in the simulation or by using different mapping functions.

To evaluate the concentration c at a set of N_g grid points, we have to find an operator $\boldsymbol{J}$ such that (Tompson and Dougherty, 1988)

$$\mathbf{Jm} \longrightarrow \mathbf{C} \tag{5.67}$$

where $\boldsymbol{C}$ is a vector of N_g concentrations, $\boldsymbol{m}$ is a vector of N_p particle masses and $\boldsymbol{J}$ is a N_g x N_p weighting matrix.

The particle mass density $\hat{C}(\boldsymbol{r},t)$ can be defined as

$$\hat{\mathrm{C}}(\mathbf{r},\mathrm{t}) = \sum_{\mathrm{p}\in \mathrm{N_p}} \mathrm{m_p}(\mathrm{t})\delta(\mathbf{r}-\mathbf{r_p}(\mathrm{t})) \tag{5.68}$$

where δ is the Dirac function, $m_p(t)$ is the mass that particle p represents at time t, and $\boldsymbol{r_p}(t)$ is the location of particle p at time t. Because the exact particle representation is discontinuous, however, smoothed approximations (5.69) have to be formed and used in its place.

$$\tilde{\mathrm{C}}(\mathbf{r},\mathrm{t}) = \int_{\mathrm{D_c}} \hat{\mathrm{C}}(\mathbf{r'},\mathrm{t})\mathrm{W_H}(\mathbf{r}-\mathbf{r'})\mathrm{d}\mathbf{r'} = \sum_{\mathrm{p}\in \mathrm{N_p}} \mathrm{m_p}(\mathrm{t})\mathrm{W_H}(\mathbf{r}-\mathbf{r_p}(\mathrm{t})) \tag{5.69}$$

where D_c is the computational domain, and W_H is a projection, weighting or regularisation function of finite support, which should satisfy the normalisation condition (5.70) for purposes of mass conservation, and ideally be invariant under coordinate transformations:

$$\int_{\mathrm{D_c}} \mathrm{W_H}(\mathbf{r})\mathrm{d}\mathbf{r} = 1 \tag{5.70}$$

Therefore, the smoothed continuous concentration distribution $\tilde{C}(\boldsymbol{r},t)$ is found by convolving the rough particle solution $\hat{C}(\boldsymbol{r},t)$ with a projection function $W_H(\boldsymbol{r},t)$.

Bagtzoglou *et al.* (1992) have investigated and compared the effects of implementing different projection functions in different flow regimes or dimensions, using concentration solutions for a set of one-, two-, and three-dimensional non-reactive test problems. The projection functions they used are the Nearest Grid Point (NGP), the Closed In Cell (CIC), the Triangular Shaped Cloud (TSC) and the Truncated Gaussian (TG) functions, which can be seen in figure (5.17) and whose defining equations are given below -(5.71) to (5.74), respectively-:

$$HW_H(x - X_p) = 1 \qquad \frac{|x - X_p|}{H} \le 0.5 \tag{5.71}$$

$$HW_H(x - X_p) = 1 - \frac{|x_g - X_p|}{H} \qquad \frac{|x - X_p|}{H} \le 1 \tag{5.72}$$

$$HW_H(x - X_p) = \begin{cases} 0.75 - \left(\frac{|x - X_p|}{H} \right)^2 & \frac{|x - X_p|}{H} \le 0.5 \\ \frac{1}{2}\left(1.5 - \frac{|x - X_p|}{H} \right)^2 & 0.5 \le \frac{|x - X_p|}{H} \le 1.5 \end{cases} \tag{5.73}$$

$$HW_H(x - X_p) = \exp\left(-\pi \left(\frac{|x - X_p|}{H} \right)^2 \right) \qquad \frac{|x - X_p|}{H} \le 1.5 \tag{5.74}$$

where now *H* represents a lengthscale.

The authors found that, to avoid an excessive smearing of the results, the support of the projection functions should be of the order of the spatial grid size. They also point out that higher-order projection functions (CIC, TSC, TG) show smaller errors than the simpler NGP method, but at an increased computational effort.

The first projection function (NGP) is equivalent to selecting a symmetric region Ξ around each grid point N_g and dividing the mass found within Ξ by its volume V_Ξ, that is

$$\tilde{C}_\Xi = m_p \frac{N_\Xi}{V_\Xi} \tag{5.75}$$

This method is referred to below (§5.2.2.1) as Box-Counting method. It is useful and has been extensively applied in the case of regular grids, but is difficult to apply when dealing with irregular grids because the definition of Ξ becomes cumbersome. In addition, even though it becomes more accurate as N_p is increased, the error decreases only with $N_p^{1/2}$. Several attempts have been done to increase the method's accuracy without the burden of adding more particles using moving averages or Fast-Fourier-Transforms (Ahlstrom et al., 1977), but these methods have proved very costly when applied to two- and three-dimensional situations.

A somewhat more advanced variation on this method is the Smoothed-Particle Hydrodynamics (SPH), or kernel algorithm (Gingold and Monaghan, 1982), which is described in detail in §5.2.2.2. In this case, the estimated concentration -equation (5.69)- is multiplied by a factor $A(\boldsymbol{r})$ that takes into account the existence of boundaries to the computational domain which limit the spatial spreading of the particles's mass.

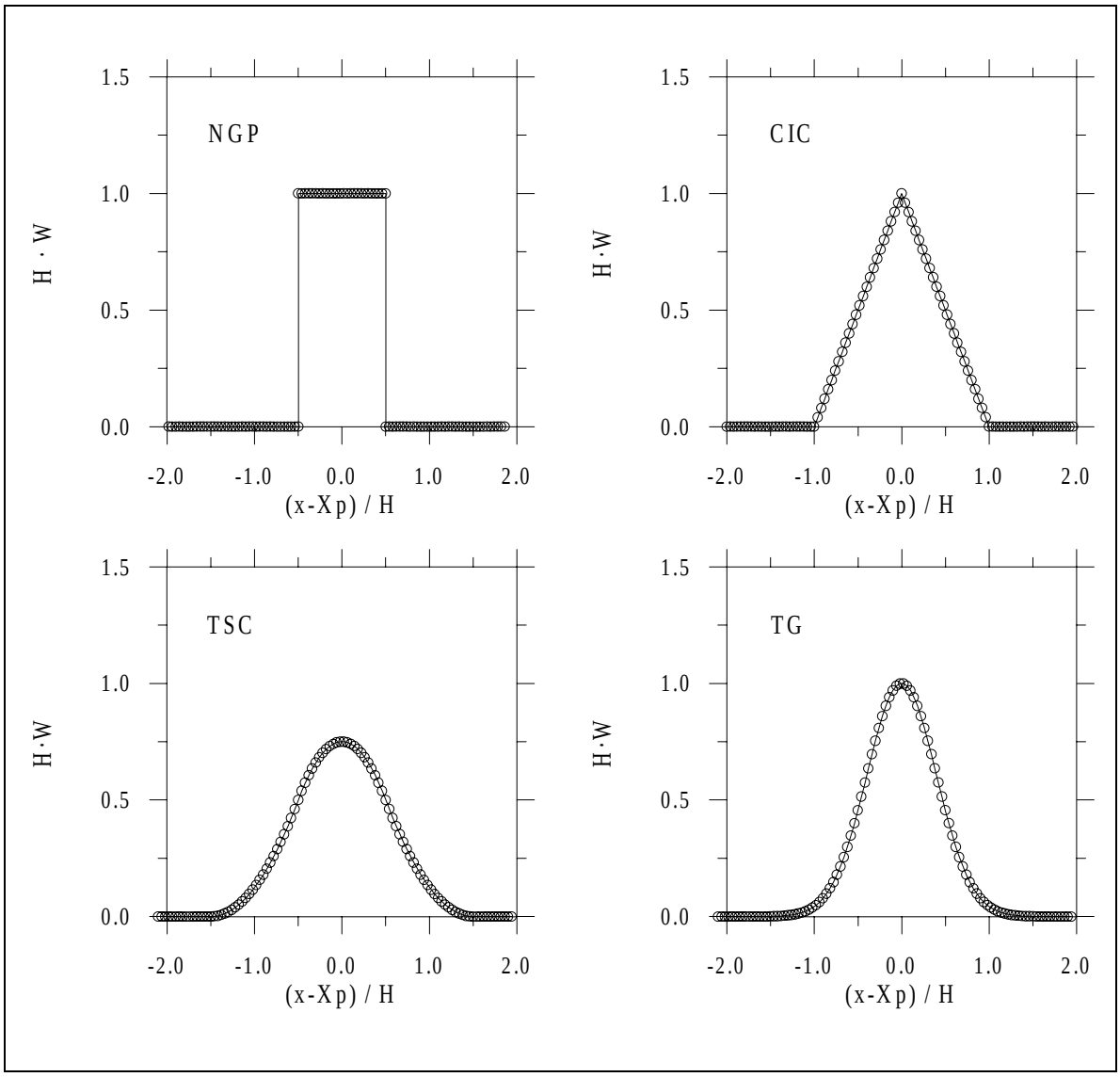

Figure 5.17: Projection functions analysed by Bagtzoglou *et al.* (1992), and defined in equations (5.71) to (5.74).

5.2.2.1 The Box-Counting method

The most direct method to obtain a continuous distribution of concentration from a discrete cloud of particles consists in overlaying a 3D concentration grid on the region occupied by the particles, counting the number of Lagrangian elements within each grid cell (CGC), adding up their masses, and dividing the total mass by the volume of the CGC. The value obtained may be assigned then to the centre of the CGC. Thus,

$$\tilde{C}_{\Xi} = \frac{1}{V_{\Xi}} \sum_{l=1}^{N_{\Xi}} m_p^l \tag{5.76}$$

where V_{Ξ} is the volume of the CGC where the concentration is to be computed, N_{Ξ} is the number of particles within said cell, and m_p^l is the mass assigned to particle l, which may be a constant, a particle characteristic, or even a time-dependent variable. In the latter case, of course, the concentration

distribution is also time-dependent. When all the particles represent the same mass, this equation takes the form of (5.75). This straightforward method allows also for the computation of concentration values at previously defined receptor points, by setting “receptor cells” around these locations and counting only which particles fall inside them.

In their model of atmospheric pollutant dispersal, however, Lamb *et al.* (1979) pointed out that this method may be too simple, and that a rigorous calculation of the concentration field should not just add up the particle masses, but should include a weighting factor related to the total time the contributing particle spent inside a given cell during each timestep.

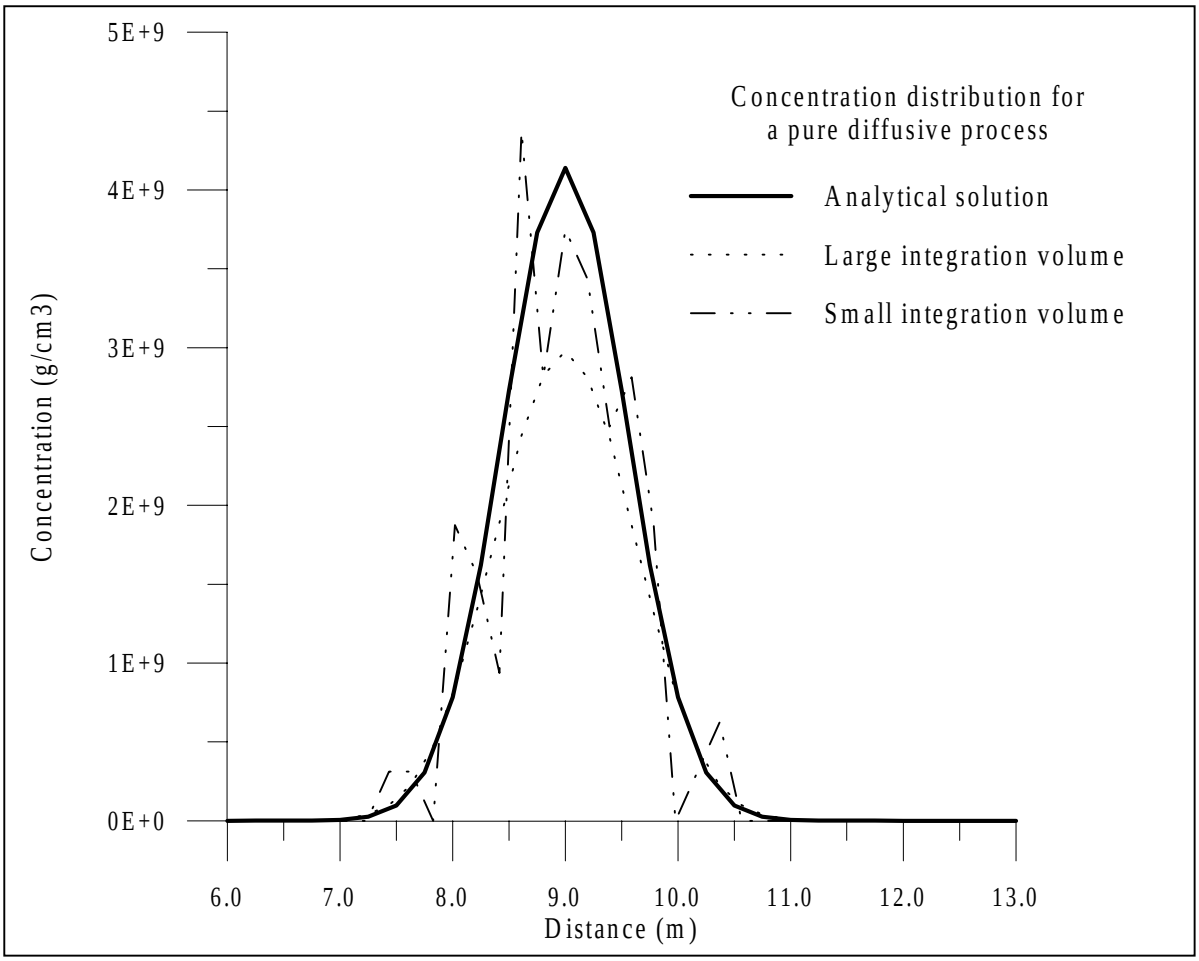

Figure 5.18: Comparison of a theoretical solution with concentration distributions obtained from a cloud of particles using a large V_Ξ (broken line) and a small V_Ξ (dotted line).

It can be seen from equation (5.76) that, given N_Ξ and m_p, the determination of concentration values from a cloud of particles using this method presents the single difficulty of finding a CGC -or integration- volume which is reliable throughout the whole space occupied by the particles, since the size of the CGC can determine in a very strong manner the accuracy of the obtained distribution. If the mass of each particle has been assigned at the beginning of the simulation, $\tilde{C}_\Xi$ depends only, but in a sensitive manner, on V_Ξ.

If the selected integration volume is very large, different points where $\tilde{C}_\Xi$ has to be evaluated may fall within the same CGC, and the resulting distribution becomes excessively smoothed out; the

limiting case of a unique concentration cell occupying the whole domain results in a constant-value concentration field. This phenomenon may occur easily around the centre of mass of the particle distribution, where the interparticle distance is small. On the other hand, the use of a very small integration volume has the opposite effect: the obtained distribution becomes severely jagged, with multiple local maxima, since it tends to reflect the individual particle positions rather than the underlying concentration field. As a result, it appears that the ideal integration volume is a function of the local particle density ψ in the sense that, where ψ is high, a small V_{Ξ} is recommended, whereas in regions of low ψ a large V_{Ξ} should be used.

This problem, which appears in some form or another in all the discrete-continuous transformation algorithms, is illustrated in figure 5.18 for the simple case of a purely diffusive process. The concentration distribution calculated with a small CGC volume, and that obtained with an excessively large CGC are compared to the theoretical Gaussian distribution.

a) The determination of the integration volume

In principle, it suffices for the integration lengthscale to be larger than the chosen resolution of the concentration field, to avoid jagged distributions. However, the numerical results could be optimised if a relationship between the integration volume and any characteristic lengthscale l_l of the most relevant mixing mechanism can be established.

For instance, we may consider the lengthscale l_l to be subject, on a physical basis, to the following conditions:

a) l_l should be smaller than, or equal to, the smallest dimension of the volume occupied by the particles,

b) l_l should be smaller than, or equal to, the smallest dimension of the computational domain (e.g., water depth for coastal regions), and

c) l_l should be smaller than, or equal to, the smallest characteristic length of the main mixing agent (e.g., wave height, turbulent eddy diameter, etc.).

The evaluation of l_l can then be done under two different criteria. If the dominant transport mechanism is advection, as is the case for the majority of natural waterbodies, the mixing will be due to stresses arising from velocity differences at nearby (shear dispersion), and the maximum distance a particle could travel during a single timestep can be approximated by

$$l_l \propto \sqrt{Kl_t} \propto \sqrt{K\frac{l_l}{l_v}} \tag{5.77}$$

from where

$$l_l \propto \frac{K}{l_v} \tag{5.78}$$

where $l_v=u_*$ is a velocity scale, and K is the “advective” dispersion mixing coefficient. On the other hand, if the pollutant is transported mainly by diffusion, then $u_*=0$ and the maximum particle displacement becomes

$$l_l \propto \sqrt{6Kl_t} \tag{5.79}$$

where now K is a "pure" diffusion mixing coefficient, and l_t is a timescale, typically the computational timestep Δt. Once an estimation of l_l has been made, the dimensions of a concentration grid cell can be determined as $\delta x_\Xi^i = \alpha^i \, l_l$.

As an example, we consider the case of a pure diffusive process, for which it can be assumed that $V_\Xi = \delta x_\Xi \cdot \delta y_\Xi \cdot \delta z_\Xi$. A set of particles were left to spread out, with a constant isotropic diffusivity K=0.01 m^2/s, for 15 seconds with a timestep Δt=1s, and then the concentration field was computed from the resulting discrete cloud, using cubic integration volumes ($\delta x_\Xi = \delta y_\Xi = \delta z_\Xi$.). Different values for V_Ξ were used, and the error between the modelled and the theoretical distribution was calculated for each one, and plotted as a function of $\delta x_\Xi{}^i = V_\Xi{}^{1/3}$. Figure 5.19 shows the existence of a value $\delta x_{\Xi\,min}^i \approx 0.35$ m which minimises the relative error; the introduction of the aforementioned values for K and Δt in equation (5.79) yields a value for the length scale of l_l =0.25 m, which corresponds to $\alpha^i \approx$ 1.5, and is quite close to the real value. However, the estimation of the adequate δ_Ξ in real cases may not be an easy task, because of the complexity of the problem.

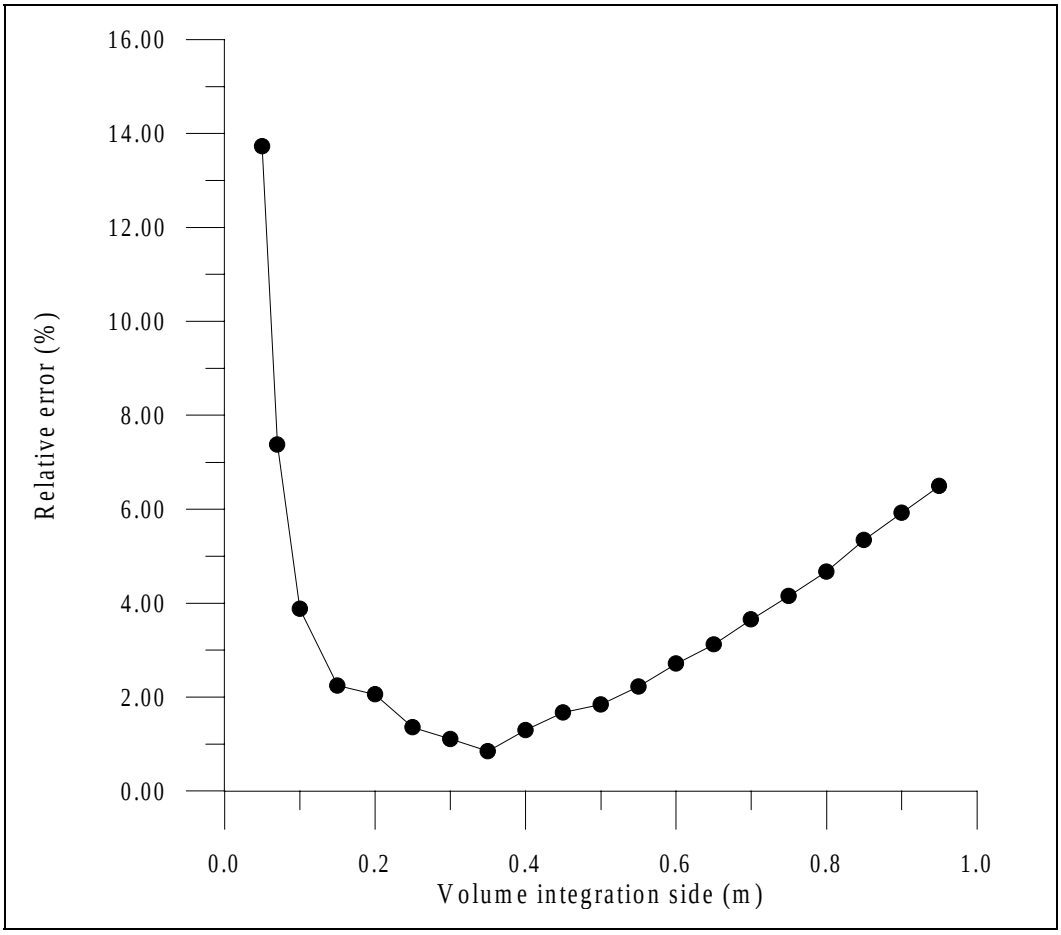

Figure 5.19: Relative error as a function of $V_\Xi{}^{1/3}$, for a purely diffusive process.

b) The kBC method

A variation may be performed on the above method in order to take into account the randomness of some of the transport processes involved in particle dispersion. A very rough approximation to the SPH algorithm (see §5.2.2.2) is accomplished by spreading out the mass of each particle between the CGC which contains it and the 26 nearest-neighbouring CGC, following a pre-

defined distribution, as depicted in figure 5.20. The mass assigned to the central cell is a fraction $N_1\%$ of the particle's mass, while the rest of the mass ($26N_2\% = 100 - N_1\%$) is equally divided between the surrounding concentration grid cells.

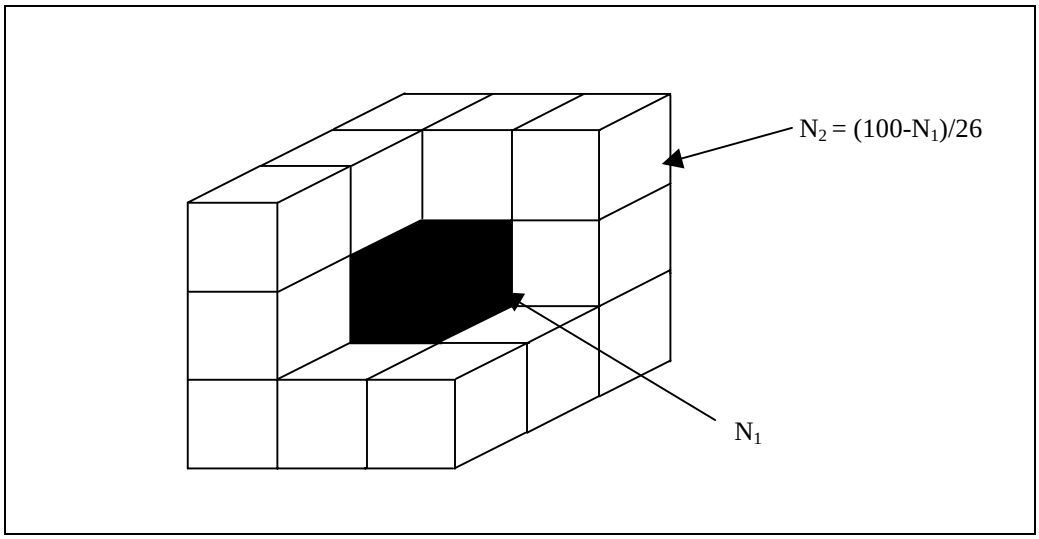

Figure 5.20: Mass distribution amongst 27 concentration grid cells in the kBC method. N_1 and N_2 (with $N_1+26N_2 = 100$) are pre-defined values.

5.2.2.2 The Smoothed Particle Hydrodynamics (kernel) method

As ordinarily defined, the concentration $\tilde{C}(\boldsymbol{r},t)$ of equal mass particles in a fluid is proportional to the number of particles per unit volume. Considered from a statistical point of view, however, it can be said that the probability of finding a particle in a volume element ΔV is proportional to $\tilde{C}\,\Delta V$. This definition can be applied to the "fluid concentration", by considering a number of discrete fluid elements; the position of these elements is regarded as a random sample from a probability density which is proportional to the mass concentration. The estimation of the concentration is then equivalent to evaluating a probability density from a sample. The statistical estimation of concentration by smoothing kernels can be interpreted as the replacement of each particle by a smoothed-out concentration ('Smoothed particle hydrodynamics', SPH) and is based on an approach first devised by Lucy (1977). This method has been successfully applied in air-quality modelling (Lorimer and Ross, 1986), in a number of astrophysical problems (see, for instance, Gingold and Monaghan, 1977) and, recently, in marine hydrodynamics (Dalrymple and Knio, 2001).

a) Introduction to kernel density estimation

As in the traditional box-counting methods, SPH algorithms employ a set of particles whose number density is proportional to a true concentration $C(\boldsymbol{r},t)$. If these particles have equal mass m_p, the concentration can be considered to be proportional either to the average number of particles per unit volume, or to the probability density of finding a particle in a given volume element. In the latter approximation, the estimate $\tilde{C}(\boldsymbol{r},t)$ of the true concentration at a position $\boldsymbol{r}$ can be written as

$$\tilde{C}(\mathbf{r},t) = \int W_H(\mathbf{r}-\mathbf{r}',H)\,C(\mathbf{r}',t)\,d\mathbf{r}' \qquad (5.80)$$

where $C(\boldsymbol{r},t)$ is the true density, H is a resolution bandwidth or smoothing length, and W_H is a smoothing kernel such that

$$\int \mathrm{W_H}(\mathbf{r},\mathrm{H})\,\mathrm{d}\mathbf{r} = 1 \tag{5.81}$$

and both integrations are over all the domain. From equation (5.81) it follows that the total mass is independent of the estimation, since

$$\mathrm{M(t)} = \int \tilde{\mathrm{C}}(\mathbf{r},\mathrm{t})\mathrm{d}\mathbf{r} = \int \mathrm{C}(\mathbf{r},\mathrm{t})\mathrm{d}\mathbf{r} \tag{5.82}$$

Because the true density is unknown, (5.80) cannot be evaluated in a direct manner. However, if a set of N_p particles exists, randomly distributed according to $C(\boldsymbol{r},t)$, the integral can be evaluated by the Monte Carlo method; defining $\tilde{C}_{N_p}(\boldsymbol{r},t)$ as

$$\tilde{\mathrm{C}}_{\mathrm{N_p}}(\mathbf{r},\mathrm{t}) = \frac{\mathrm{M}}{\mathrm{N_p}} \sum_{\mathrm{j=1}}^{\mathrm{N_p}} \mathrm{W_H}(\mathbf{r}-\mathbf{r_j},\mathrm{H}) \tag{5.83}$$

we find that

$$\mathrm{E}\left[\tilde{\mathrm{C}}_{\mathrm{N_p}}(\mathbf{r},\mathrm{t})\right] = \frac{1}{\mathrm{M}^{\mathrm{N_p}}} \int \ldots \int \tilde{\mathrm{C}}_{\mathrm{N_p}}(\mathbf{r},\mathrm{t}) \prod_{\mathrm{i=1}}^{\mathrm{N_p}} \mathrm{C}(\mathbf{r_i})\mathrm{d}\mathbf{r}_\mathrm{i} = \tilde{\mathrm{C}}(\mathbf{r},\mathrm{t}) \tag{5.84}$$

if H is independent of the sample, and where E denotes expectation. This expression should be interpreted in the sense that, if we were to have an ensemble of different sets of points consistent with the initial conditions, the ensemble average of $\tilde{C}_{N_p}(\boldsymbol{r},t)$ would be $\tilde{C}(\boldsymbol{r},t)$.

In situations in which large changes in the density occur, the smoothing length must be related to the ensemble of particles, and equation (5.84) is approximately true with H replaced by its expected value. The only requirement to be imposed on H is that $H \to 0$ as $N_p \to \infty$, so that $\tilde{C}(\boldsymbol{r},t) \to C(\boldsymbol{r},t)$; statistically, the dependence of H on N_p is limited by the need to keep fluctuations from growing too quickly, and it is convenient to take

$$\mathrm{H} \propto \mathrm{N_p^{-d}} \tag{5.85}$$

where d is the number of dimensions (Gingold and Monaghan, 1982).

The use of the simple density estimator (5.83) presumes that the mass of each particle is distributed smoothly over an unbounded space. When a particle cloud is generated within a bounded domain, (5.83) is not able to model concentration values correctly at the boundaries, and the estimate must be modified by introducing a factor

$$\mathrm{A}(\mathbf{r}) = \frac{1}{\int_{\mathrm{D_c}} \mathrm{W_H}(\mathbf{r}-\mathbf{r}',\mathrm{H})\mathrm{d}\mathbf{r}'} \tag{5.86}$$

so that, if m_p is the mass of each particle, equation (5.83) becomes

$$\tilde{C}_{Np}(\mathbf{r}) = A(\mathbf{r}) m_p \sum_{j=1}^{N} W_H(\mathbf{r} - \mathbf{r}_j, H) \tag{5.87}$$

The evaluation of $A(\boldsymbol{r})$ is straightforward for planar domains; when these are irregular, the integral in (5.86) may become difficult to evaluate, and terrain-following coordinates (Lorimer, 1986) must be introduced to solve the problem.

It is convenient, for future purposes, to write the smoothing kernel in the following form, assuming a three-dimensional space:

$$W_H(\mathbf{r} - \mathbf{r_i}, H) \equiv \frac{1}{H^3} W_H\left(\frac{\mathbf{r} - \mathbf{r_i}}{H}\right) = \frac{1}{H^3} W_H(\mathbf{u}) \tag{5.88}$$

As with the BC method, it is not enough to simply choose a mapping algorithm, but one must also select values or expressions to assign to some free parameters. In this case, both the smoothing length *H* and the smoothing kernel function W_H must be determined *a priori*, and the model results will depend on the validity of this choice.

b) Selecting the smoothing kernel

There are a number of conditions that a function must satisfy to be eligible as a smoothing kernel. In the first place, and since in SPH methods no grid is defined and the concentration gradients are computed when necessary by differentiating *C*, the smoothing kernel must be differentiable at least once, and its derivative should be continuous in order to prevent large fluctuations in the calculations.

It is also advisable to choose the smoothing kernel as a member of a sequence of functions which approximates a δ function. This is justified by noting that, if W_H were a true delta function, the estimated density evaluated with equation (5.80) would be equal to the true density.

A further restriction on the form of the kernel may be set by considering the bias, defined as the difference between the estimated and the true density at a certain point:

$$\beta = \tilde{C}(\mathbf{r}) - C(\mathbf{r}) \tag{5.89}$$

If *W* has a sharp peak at the position $\boldsymbol{r} = \boldsymbol{r'}$, the integrand of (5.80) can be expanded about $\boldsymbol{r'} = \boldsymbol{r}$, yielding

$$\beta = \nabla C \int \mathbf{u} W_H(\mathbf{u}) d\mathbf{u} + \frac{1}{2} \int [(\mathbf{u}\nabla)(\mathbf{u}\nabla C)] W_H(\mathbf{u}) d\mathbf{u} + \ldots \tag{5.90}$$

The first term in the bias is zero if W_H is an even function, and the important term in β becomes

$$\frac{1}{6} \nabla^2 C \int u^2 W_H(\mathbf{u}) d\mathbf{u} \propto H^2 \nabla^2 C \tag{5.91}$$

which can be removed by choosing an appropriate form of W_H.

Typical examples of kernel functions used in three-dimensional particle modelling are presented below. Any function that complies to the aforementioned conditions can be used, however, as a smoothing kernel:

$$\left(\frac{1}{\pi H^2}\right)^{3/2} \exp(-u^2) \tag{5.92a}$$

$$\frac{3\mathscr{H}(1-u)}{4\pi H^3} \tag{5.92b}$$

$$\frac{10}{7\pi}\frac{1}{H^3}\begin{cases}\left(1-\frac{3}{2}u+\frac{3}{4}u^3\right) & 0\le u\le 1\\ \frac{1}{4}(2-u)^3 & 1\le u\le 2\\ 0 & \text{elsewhere}\end{cases} \tag{5.92c}$$

where $\mathscr{H}$ is the Heaviside step function. The second function does not have a continuous derivative, and is thus not useful when gradient calculations are required (Monaghan, 1985). The third kernel is a member of the basic splines group (Monaghan, 1985), which have a compact support (i.e., for $r > 2H$ the interactions are exactly zero), thus limiting the number of particles to be considered in the calculations.

c) Selecting the smoothing length

The bandwidth H is a length which determines the degree of overlap between the particles's individual contribution to the concentration value at a certain point $\boldsymbol{r}$; only particles at a distance from $\boldsymbol{r}$ equal to, or smaller than, $2H$ make a significant contribution to $\tilde{C}(\boldsymbol{r},t)$.

The importance of the smoothing length lies in the fact that its value can affect in a great measure the results of a simulation; it can be shown that, for two kernels with similar analytical properties, the smoothness and accuracy of the estimates depend primarily on the bandwidth H. If H is much smaller than the typical interparticle distance l_p, the concentration distribution becomes "jagged", reflecting the particle positions rather than the underlying probability density; it is said that the estimated concentration distribution suffers from excessive variance. If, on the other hand, H is much greater than l_p, the distribution becomes overly smoothed, and it is said to suffer excessive bias. Therefore, it is necessary to find a bandwidth that balances variance and bias, and this is done by relating it somehow to a natural length scale, such as the mean interparticle distance in the sample.

In special situations where extreme conditions may generate very sharp concentration gradients, keeping a constant value for H throughout the whole domain results in a loss of accuracy in the solution, because of the wide range of interparticle distances which lead to smoothing in some regions, and jagging in others. The straightforward solution is to introduce local values for H at each particle position, related somehow to the local mean interparticle separation. This assignment of individual bandwidths presents a series of advantages, such as reducing the dependence of the computed error on the choice of the different parameters, and a closer approximation to real values, but it also burdens the model with an increase of the required computational time, due both to the greater complexity of the involved equations and to the large number of H values that must be computed. Nevertheless, it can be shown that to obtain qualitatively similar results, SPH methods

require a considerably smaller number of particles than traditional Box-counting methods do, and perhaps the excess time invested by SPH in the transformation process is balanced by the smaller CPU time employed in calculating particle positions.

5.2.3 Uncoupling the nearfield and the farfield

The behaviour of a pollutant near the discharge point is quite different from what it is farther away from it. As was mentioned before (§4.3), in the early stages of transport the dispersion of a substance is governed mainly by discharge parameters, such as the source configuration, the discharge velocity, and the effluent density. As time goes by, the effluent mixes with the ambient water, and the initial effects tend to vanish compared to the effects of such environmental factors as hydrodynamic currents and ambient turbulence.

The time-evolution of a jet (or plume) in the nearfield is generally characterised by a continuous velocity decrease, and possibly by changes in the jet (or plume) trajectory. In the case of a jet, the former is because the initial discharge velocity is usually larger than the hydrodynamic currents, and the turbulence induced by velocity shear at the jet's interface tends to entrain slower ambient water into the pollutant, thus decreasing its overall velocity; the trajectory variations depend on the capability of the ambient currents to deflect the jet. In the case of a plume, the entrainment of "clean" water into the body of the plume decreases the density difference with respect to the ambient water density, thus reducing the buoyancy forces acting on the pollutant and slowing down the plume rise; as the rise velocity decreases, the ambient currents also deflect the plume trajectory.

These rapid variations in the jet (or plume) characteristics, together with the fact that vertical transport in the nearfield is very important, require the definition of a small computational timestep in order to correctly reproduce the nearfield transport and obtain coherent concentration distributions.

When the simulation extends to the farfield transport, however, variations in the characteristics of the pollutant field are not so fast, since the effluent trajectory is determined by the hydrodynamic field. In addition, buoyancy effects may be neglected, and the transport becomes predominantly horizontal. Under these circumstances, the small computational timestep required for the nearfield simulation becomes a burden, considerably increasing the total number of particles and total CPU-time required, but without offering an equivalent increase in accuracy.

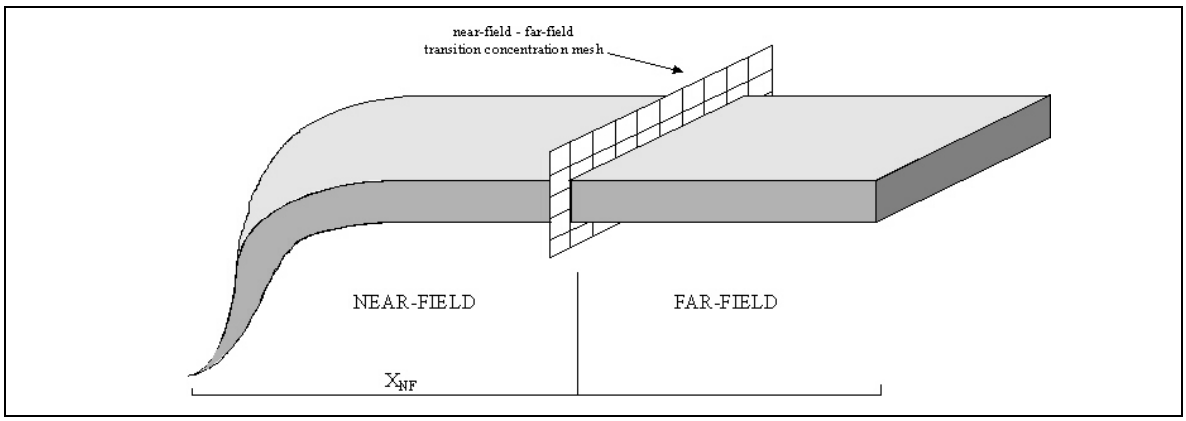

Figure 5.21: Schematic representation of the near- and farfield zones, with the vertical concentration mesh used to uncouple both.

Therefore, the possibility of uncoupling the simulation of nearfield and farfield transport is offered. In this scheme, two consecutive model runs are required, one for each region. The first run is performed using the initial discharge parameters, and stops automatically when the first batch of particles reaches the end of the nearfield, given by the asymptotic expression proposed by Huang *et al.* (1996), after the work of Wright (1977) and Doneker and Jirka (1990):

$$x_{NF} = \begin{cases} x_{NF}^{(1)} & H_d / l_B < 0.1 \\ C_1 x_{NF}^{(1)} + C_2 x_{NF}^{(2)} & 0.1 \le H_d / l_B < 10 \\ x_{NF}^{(2)} & 10 < H_d / l_B \end{cases} \tag{5.93}$$

where x_{NF} is the downstream distance at which the nearfield ends, l_b is the initial buoyancy lengthscale defined in (4.99d), H_d is the discharge depth, and the rest of variables are defined in the following equations:

$$x_{NF}^{(1)} = 0.5824 \frac{H_d^{4/3}}{l_B^{1/3}} + 3H_d \tag{5.94}$$

$$x_{NF}^{(2)} = 1.0 \frac{H_d^{3/2}}{l_B^{1/2}} + 0.6H_d \tag{5.95}$$

$$\begin{aligned} C_1 &= 0.5 - 0.5 \log\left(\frac{H_d}{l_B}\right) \\ C_2 &= 0.5 + 0.5 \log\left(\frac{H_d}{l_B}\right) \end{aligned} \tag{5.96}$$

The set of particles resulting from this first run is mapped onto the nodes of a vertical computational mesh (figure 5.21).using one of the mapping algorithms described above. Each cell of the mesh is then defined to be a “virtual source”, of computed concentration, for the second model run, which is limited to the farfield, and may be done using a larger timestep than for the previous (nearfield) run

5.2.4 Model flowchart

The LIMMIX model consists of two separate modules. The first one computes the dispersion of a set of particles, whereas the second transforms the resulting discrete particle cloud into a continuous distribution of concentrations.

5.2.4.1 The transport module

The basic functioning of the dispersion module is summarised in the flowchart given in figure 5.22, and will be here briefly described.

The setup of the case to be modelled is performed by reading the setup files containing data such as the dimensions of the computational grid (number of cells and cell size), the characteristics of the discharge and the effluent (orientation, position, time-variability, initial concentration, initial volume flux, density, etc) and environmental parameters (water density profile, presence of wave

field, etc). The setup file also contains the names of the files in which the bathymetry and the boundary reflectivity indices are stored, and the files rendered by a hydrodynamic model which contain relevant hydrodynamic data (velocity components, stress components, etc.). Numerical data such as initial and final simulation time, computational timestep, number of particles to be used, type of mapping method to be used, type of output results to be given, etc, are also included in this file.

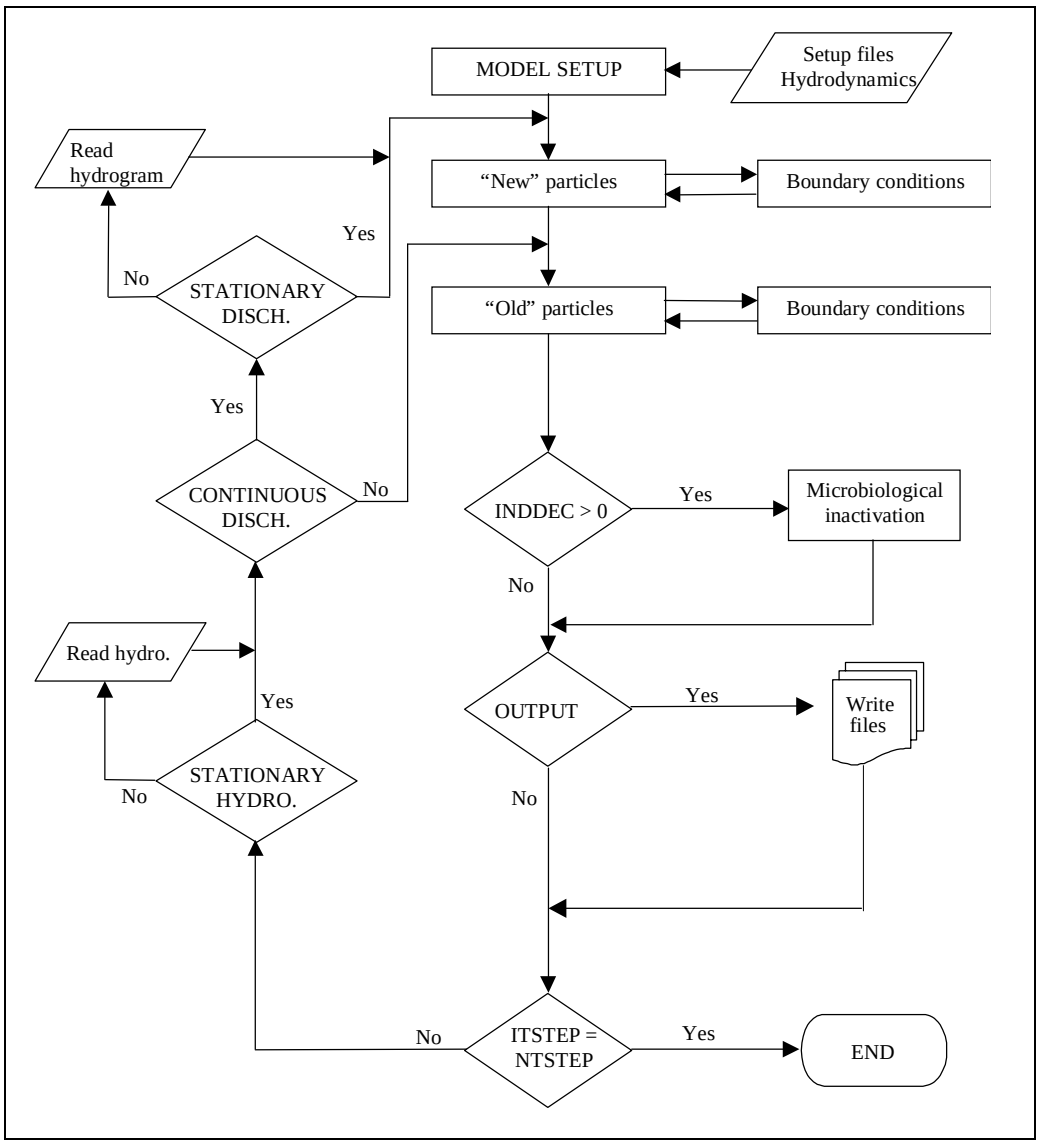

Figure 5.22: Flowchart of the LIMMIX dispersion model.

Once the case setup has been done, the first batch of particles, termed "new" in the flowchart, are moved according to equations (5.15), after computing the components of the "pseudovelocity", and boundary conditions are duly applied. If the effluent contains bacteria, the microbiological inactivation module will then be activated.

The procedure outlined in the previous paragraph is repeated as many times as determined by the total computation time and the timestep, taking into account if the hydrodynamics is non-stationary and if the discharge is non-stationary and continuous. In the former case, updated hydrodynamic files are read; in the second, the corresponding hydrograms are read. If the discharge is

continuous, at each timestep there will be a set of “old” particles to be moved after the “new” particles.

When it is time to output the results, either at the end of the simulation time or at a fixed time interval during the model run, the positions of all particles are printed into files. The particle positions are considered to be the fundamental result from the dispersion model, and may be used as input data for the transformation module. In this manner, the discrete-continuous transformation can be repeated as many times as necessary, refining the concentration grid parameters as required.

5.2.4.2 The mapping module

The concentration module setup is done in a similar way, reading an input data file which specifies model parameters (type of transformation algorithm used, size of the CGC, resolution of the concentration grid, etc.), bathymetric and hydrodynamic files, and the file containing the particle positions. Depending on the selected transformation method, the model will call routines KBC, SPH or PARTCON, which will compute the concentration distribution using a grid-based BC method, or the SPH or BC method centred at the position of each particle. When this is completed, the results will be printed in a file, and the program will end. Variable IFIELD = 2 states that the output data will be used as initial data for a farfield run of the dispersion model, and therefore only a grid-based algorithm should be used, printing the results out in a special format (N_POS_F).

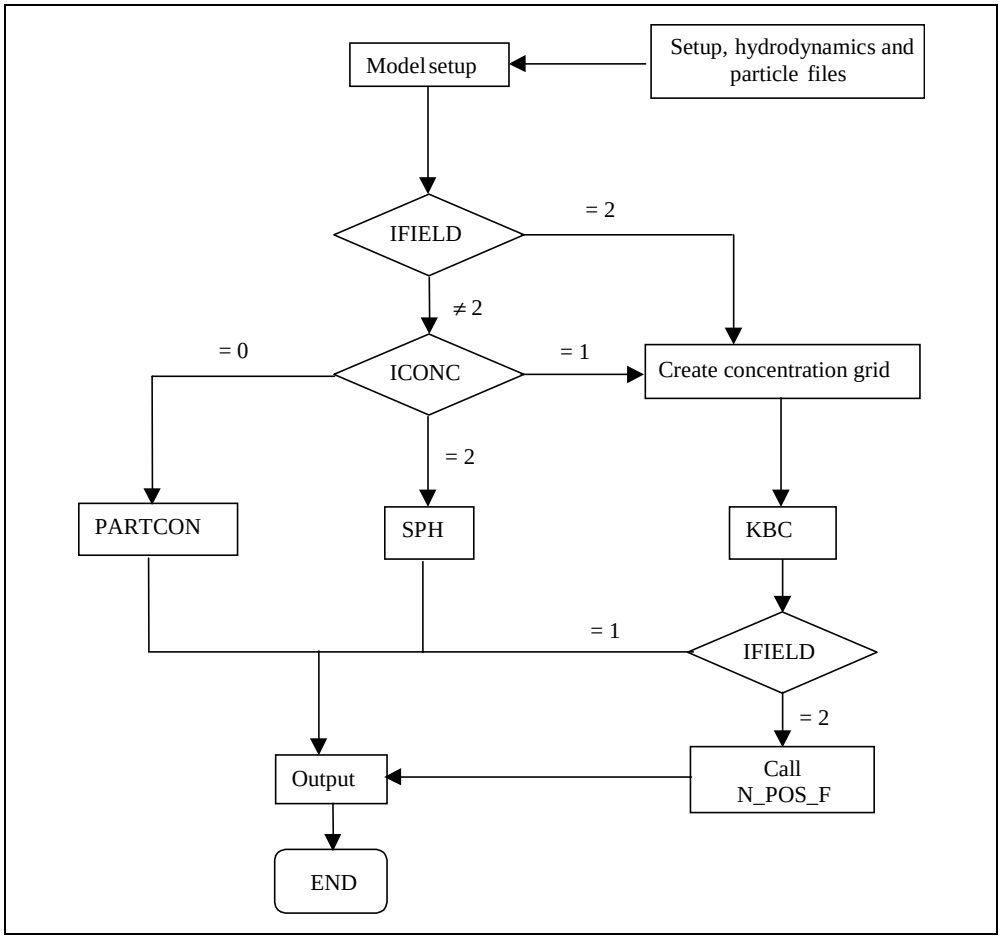

Figure 5.23: Flowchart of the LIMMIX mapping module.

5.3 MODEL VALIDATION

In order to validate different aspects of the transport model, simple cases of known analytical solution or available experimental data have been selected. In particular, the bacteria inactivation module was checked by comparing the modelled solar radiation with field measurements, and the number of particles as a function of time with a theoretical expression; the random walk algorithm used to model the diffusive transport was verified using the case of a 3D discharge from a point source in a stagnant environment, and the combined effect of advective and diffusive transport was validated by considering a substance discharge in a one-dimensional flow field. The latter two cases can be described by analytical functions which may be found in the literature, and have also been used to estimate the performance of the three particle-node mapping functions described above.

5.3.1 Boundary conditions

A simple scenario was set up to check the validity of the formulation referring to the solid boundary conditions. The main goal of the test was to confirm that "dry" cells" were inaccessible to particles, and also that the transport of the latter through the sea bed was impossible.

In the test, it was assumed that a continuous source of particles was located at the horizontal centre (x_0 = 5 m, y_0 = 2 m) of the computational domain illustrated below, and at 1 m depth.

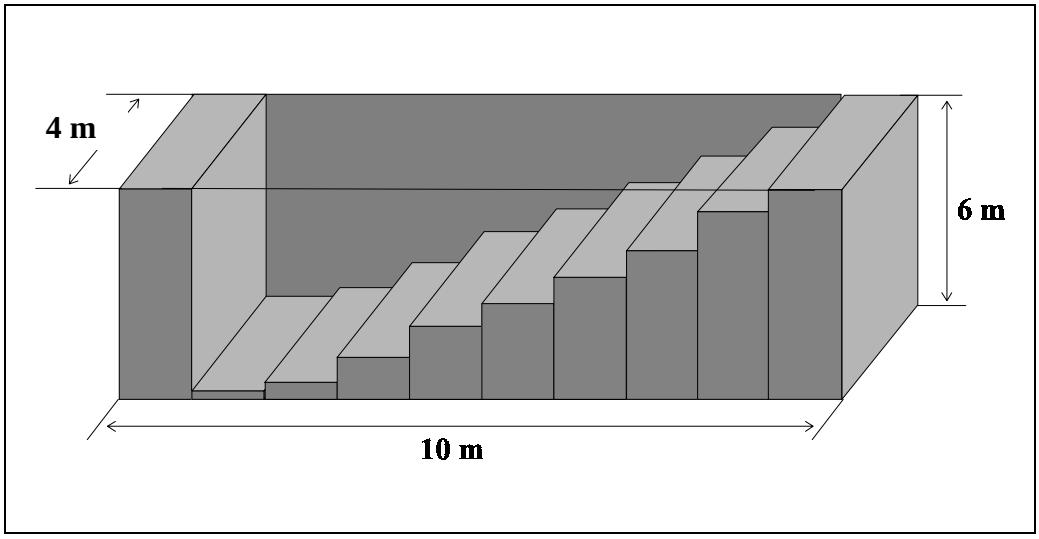

Figure 5.24: Computational domain used to test the formulation of boundary conditions in the numerical model.

The displacement of the particles was due only to diffusion, with diffusivities defined as D_x = 0.13 m^2/s, D_y = 0.11 m^2/s, and D_z = 0.11 m^2/s; since the modelled situation does not correspond to any real case, these values are not too important. The total simulation time was of 50 seconds, with a computational timestep of 0.25 s. The total number of particles released during this time period was 20,100, and their positions at the end of the model run are shown in figures (5.25) and (5.26). The former plots the projection of the particles's location on the *xy* plane, whereas the latter is the projection on the *xz* plane; the thick solid line marks the boundaries of the domain:

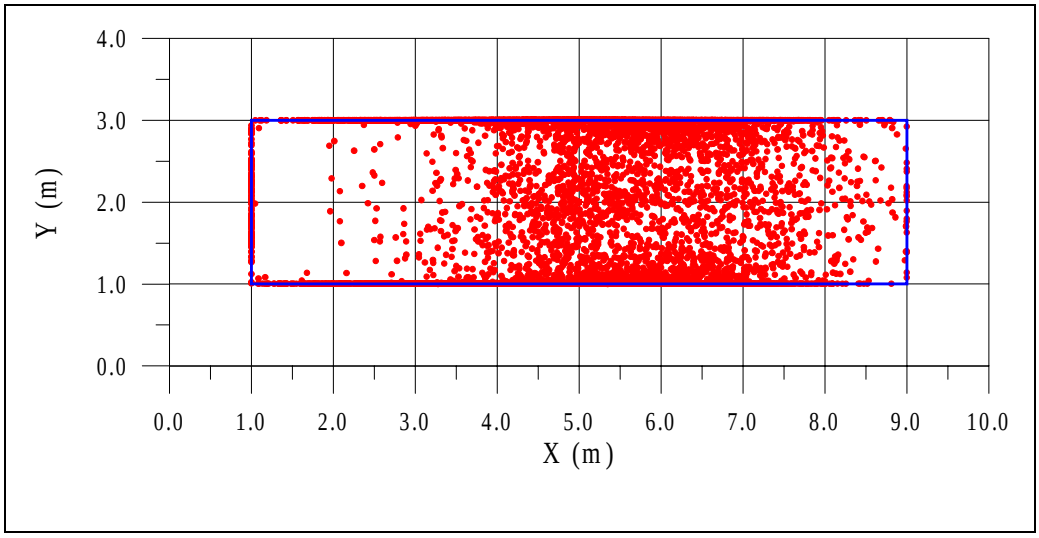

Figure 5.25: Projection on the *xy* plane of the particles's positions at the end of the model run. The solid line defines the boundaries of the physical domain.

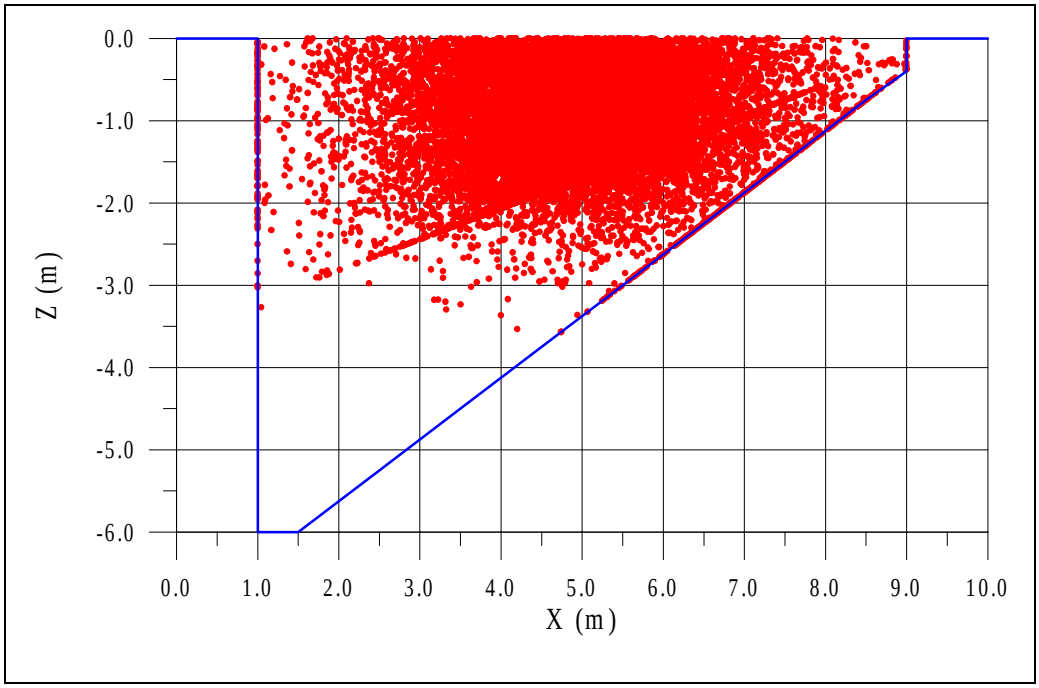

Figure 5.26: Projection on the *xz* plane of the particles's positions at the end of the model run. The solid line defines the boundaries of the physical domain, after depth interpolation.

It can be seen that the particles become trapped by the boundaries of the domain, as it was expected since the boundary surfaces were defined as non-reflective. The off-centre distribution of particles in figure 5.25 is due to the fact that, for $x < 5$, the particles reach the lateral boundaries $y = 1$ and $y = 3$ easier, whereas for $x < 5$ they tend to become immobilised at the bottom surface, all across the width of the basin. The vertical distribution of particles shows a distribution centred approximately at the discharge point, and limited by the solid boundaries. The line of particles starting at $(x, z) \cong (1.5,-3)$ and going up to $(x, z) \cong (9,-0.5)$ is due to the interpolation procedure used to smooth the topography of the

domain, avoiding sharp depth variations; therefore, the lateral walls of the domain are not vertical, but slightly inclined.

This test, although simple, permits to assess the validity of the boundary conditions as formulated in the model, since it shows that the particle transport respects the presence of solid boundaries. Other tests referring to open boundaries, or the combination of solid and open boundaries, were also done, although they are not shown here.

5.3.2 Bacteria decay module

The module corresponding to the computation of bacteria inactivation has been checked, firstly, in order to be sure that the modelling of solar intensity is accurate and, secondly, to confirm bacterial decay by monitoring the number of particles which remain in the simulation as a function of T_{90}.

The solar radiation has been modelled using equation (3.11), with the coefficients given by Colomer and Casamitjana (1994), and the results –figure 5.27c – have been compared to the experimental measurements –figs. 5.27a,b– obtained by these authors on two different days (August 29th, 1989 and February 22nd, 1990). As it can be seen, modelled and experimental values appear to agree very well.

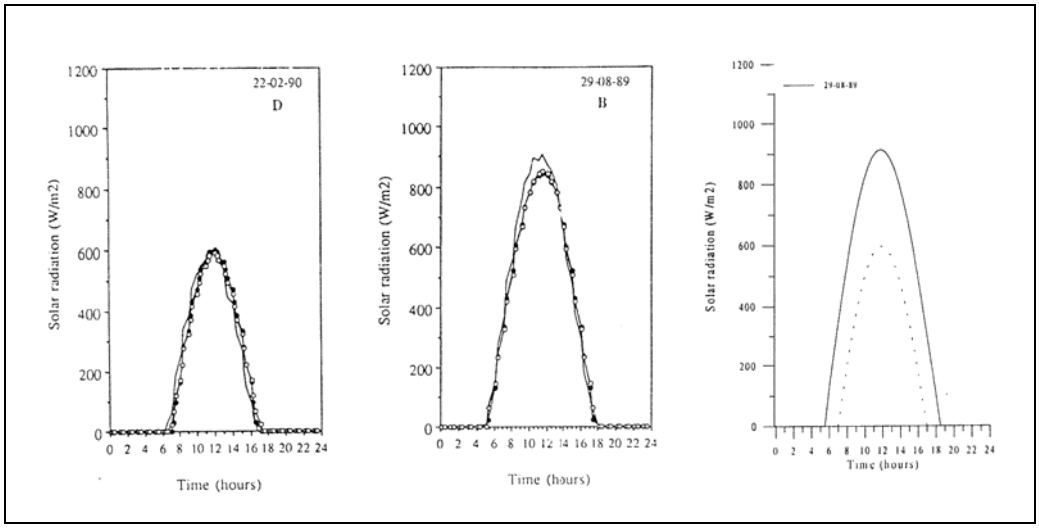

Figure 5.27: a), b) Values of the solar radiation at the surface of the Lake of Banyoles, measured by Colomer and Casamitjana (1994), and **c)** Modelled intensity.

On the other hand, the number of particles which remain in the computations after the inactivation module is applied has been calculated and compared with the theoretical values. If the particle disappearance is assumed to be exponential (i.e., exponential inactivation pattern), the number of Lagrangian elements which will remain in the simulation at a time $n\Delta t$ after the beginning of a continuous –uniform– discharge can be shown to be

$$N_n = N_0 e^{-K_d \Delta t} \frac{1 - e^{-K_d \Delta t n}}{1 - e^{-K_d \Delta t}} \tag{5.97}$$

where N_0 is the number of particles discharged in a single timestep.

Three model runs have been performed, using the same discharge parameters, but varying the inactivation coefficient. The evolution of a continuous discharge was simulated during a 90-second period, with a 3-second timestep, and an input of 10 particles/timestep; the inactivation coefficients were chosen to be T_{90} = 1 min, 4 min, and infinite. The results are shown in figure 5.28, compared with the corresponding values given by equation (5.97). It can be seen that the agreement is very good for all cases, and that the modelled curve reproduces well the behaviour of the theoretical function for all *n*.

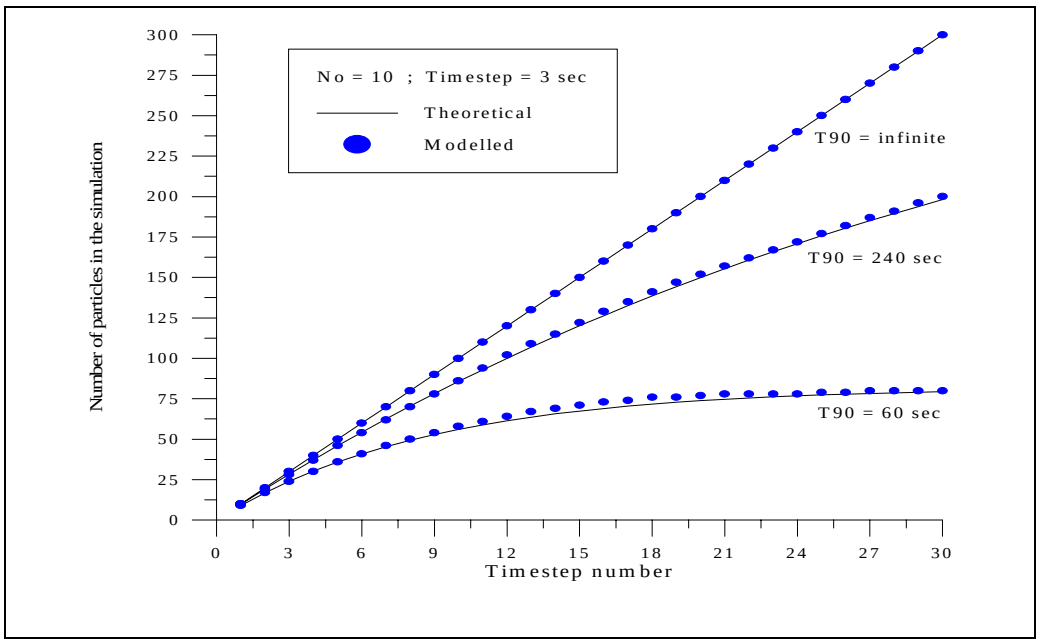

Figure 5.28: Modelled and theoretical number of particles remaining in the computations, for different values of T_{90}, after the inactivation module has been applied.

5.3.3 Three-dimensional pure diffusion of an instantaneous discharge

When a pollutant is released in stagnant environments, the dispersion and mixing with the surrounding fluid is caused exclusively by density-induced vertical motion and by molecular diffusion; if the density of the discharged substance is equal to that of the ambient fluid, molecular diffusion becomes the unique dispersion mechanism. Although this situation may not appear to be too realistic, it provides a useful test case to check the behaviour of the random number generator, estimate the validity of the random walk module implemented in the model, and compare the accuracy of the different mapping methods introduced in section 5.2.2.

Under the above assumptions, the 3D transport equation may be expressed as

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + D_z \frac{\partial^2 C}{\partial z^2} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} \right) \qquad (5.98)$$

where isotropic diffusion has been considered. If a mass M is deposited at (x_0, y_0, z_0) at time t_0=0, in a three-dimensional fluid, the resulting concentration distribution is given by (Fischer *et al.*, 1979):

$$C(x,y,z,t) = \frac{M}{(4\pi tD)^{3/2}} \exp\left[-\frac{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}{4tD} \right] \tag{5.99}$$

An instantaneous release of a conservative pollutant was modelled using the developed numerical code, and the results –continuous concentration distributions found using all three of the mapping methods described above– were compared with the analytical solution (5.99). It was assumed that a set of 110,000 particles was introduced in a stagnant fluid at position (9,9,10), and let to diffuse for a short period of 15 seconds, after which the concentration profiles were computed. The diffusion coefficient was taken to be D = 0.1 m^2/s, and the computational timestep was Δt = 1 s. The proposed situation is not realistic, since D is several orders of magnitude larger than real molecular diffusivity, but it does allow to validate several features of the transport model.

To analyse the results obtained, only the concentration distribution along the three coordinate axes was considered, and the error measured along these principal axes was assumed to be an estimate of the total error. The relative error function along the axis i was defined as

$$\varepsilon_i = \frac{100}{N^i_{grid}} \sum_{j=1}^{N^i_{grid}} \frac{\left| C^T_{ij} - C^C_{ij} \right|}{C^T_M} \tag{5.100}$$

where C_{ij}^T and C_{ij}^C represent the theoretical and the computed concentration, respectively, at evaluation point (i, j), C_M^T is the maximum value of the theoretical concentration, and N^i_{grid} is the number of evaluation point along the axis x_i; the total relative error ε_t was calculated as the mean value of (5.100) along all three axes.

Several tests were carried out using increasing numbers of particles, to study the influence of N on the accuracy of the solution. Both the BC and the Smoothed-Particle Hydrodynamics methods were used to transform the discrete particle distribution into a continuous concentration distribution, and the error with respect to the analytical solution was calculated. Figures 5.29, 5.30 and 5.31 show the concentration profile (in arbitrary units) obtained using a set of 100,000 particles, with a cubic integration volume $V_\Xi = l_\Xi^3$, where l_Ξ lies between 0.05m and 0.95m; the upper limit was selected after plotting the relative error ε_t *versus* l_Ξ, and observing that a minimum occurred at around l_Ξ=0.35 (figure 5.32), increasing linearly for larger values of l_Ξ. All of the concentration profiles have been computed using the BC algorithm described above.

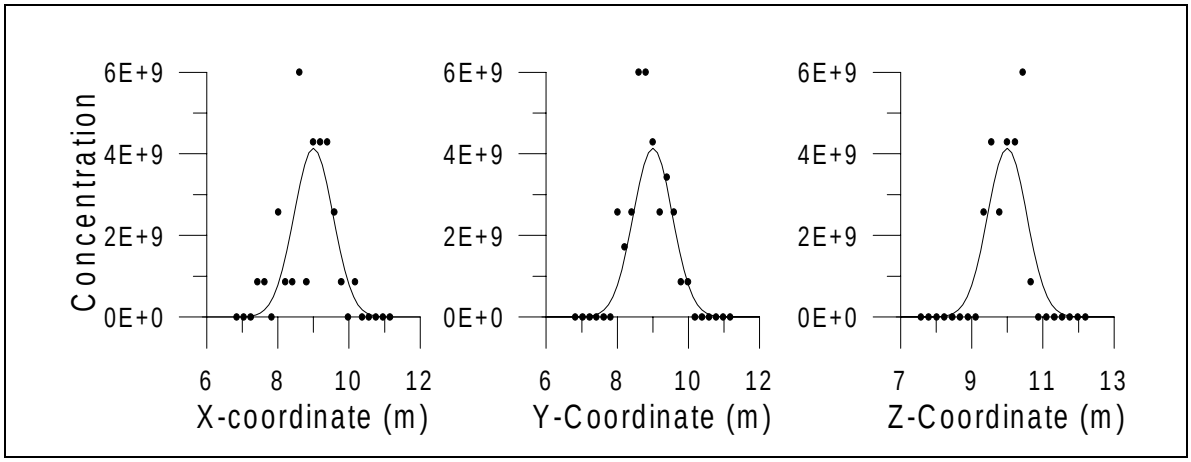

Figure 5.29: Concentration distribution (arbitrary units) calculated for a set of 100,000 particles, using BC method and an integration volume edge l_{Ξ} = 0.05 m.

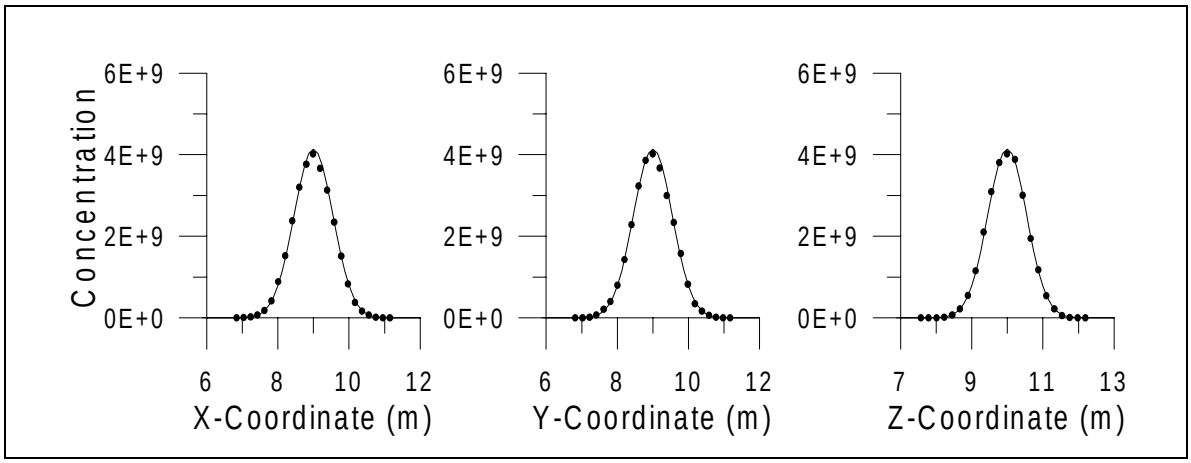

Figure 5.30: Concentration distribution (arbitrary units) calculated for a set of 100,000 particles, using BC method and an integration volume edge l_{Ξ} = 0.35 m.

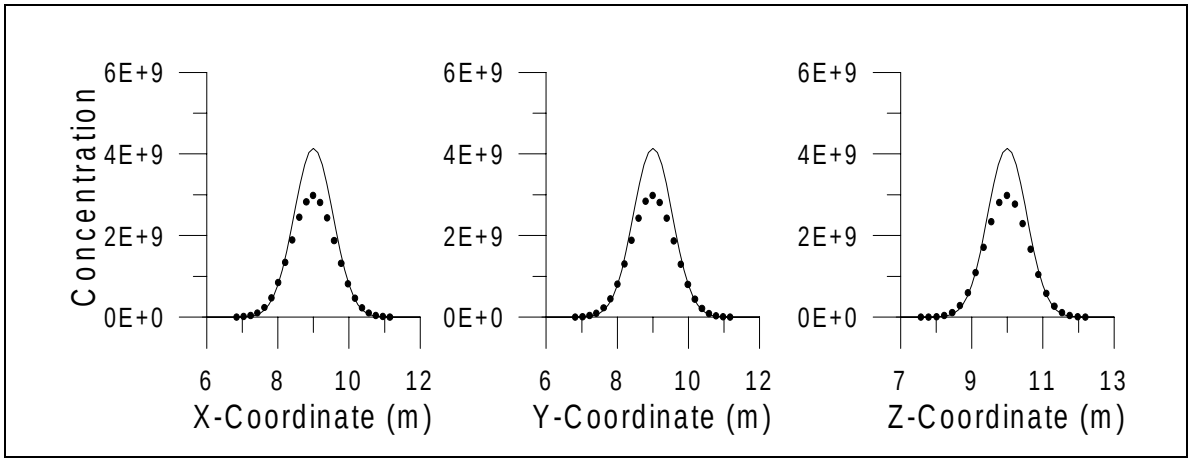

Figure 5.31: Concentration distribution (arbitrary units) calculated for a set of 100,000 particles, using BC method and an integration volume edge l_{Ξ} = 0.95 m.

The comparison of these graphs also illustrates the effects of choosing an integration lengthscale which is too small (jagged distributions) or too large (oversmoothed distributions), as was previously mentioned (figure 5.18).

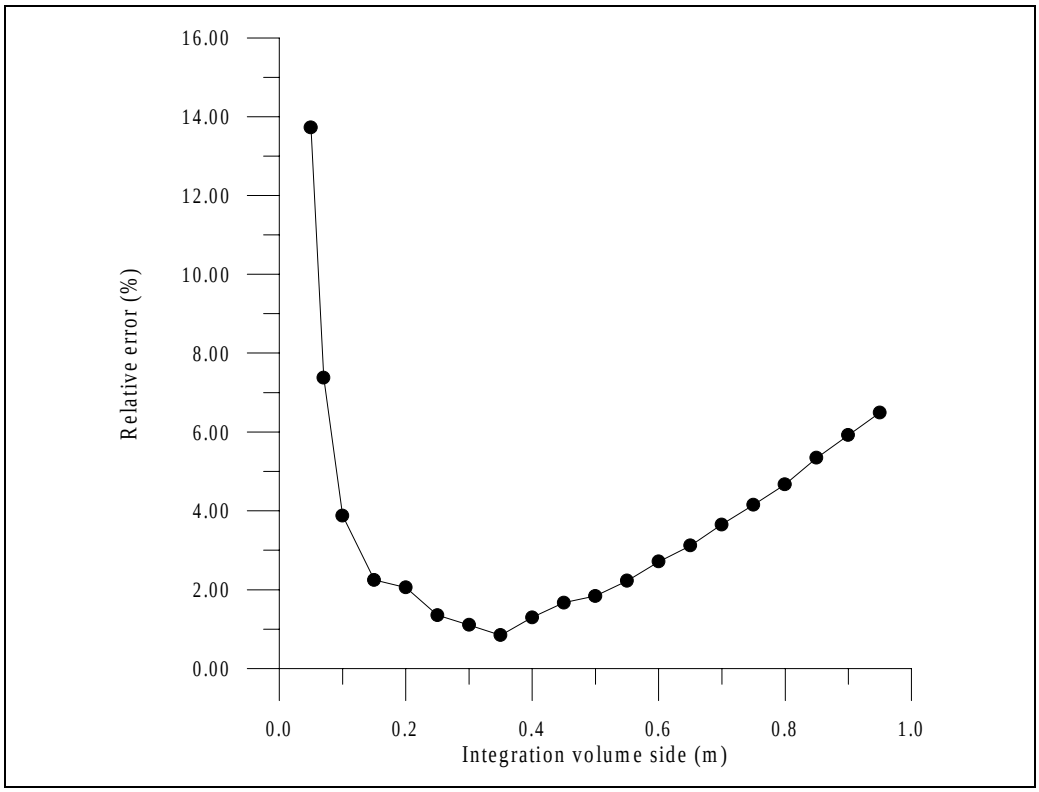

Figure 5.32: Relative error as a function of the edge of the integration volume, using BC method, and for a set of 100,000 particles.

A further analysis was done in order to determine the behaviour of the relative error as a function of the number of released particles. Based on the results shown in figure 5.32, the edge of the integration volume was taken as 0.35m, in order to minimise the modelling error, and the concentration distribution and the associated error were computed using between 5,000 and 100,000 particles.

Some of the concentration profiles that were obtained are shown below, while the calculated relative error is shown in figure 5.36. It can be seen that there is a particle number after which an increase in N and, therefore, in computation time, can not be justified by the small increase in accuracy that is found. This threshold is expected to appear in all other problems solved using a particle model, although its particular value will be problem-dependent, and also depend on numerical parameters of the simulation, on the mapping algorithm used to obtain continuous distributions, and on the integration length (or volume) defined. Therefore, although in this case, it is seen from figure 5.36 that the threshold appears to be between N = 35,000 and N = 40,000 particles, it is not possible to define an absolute value valid for all simulated cases.

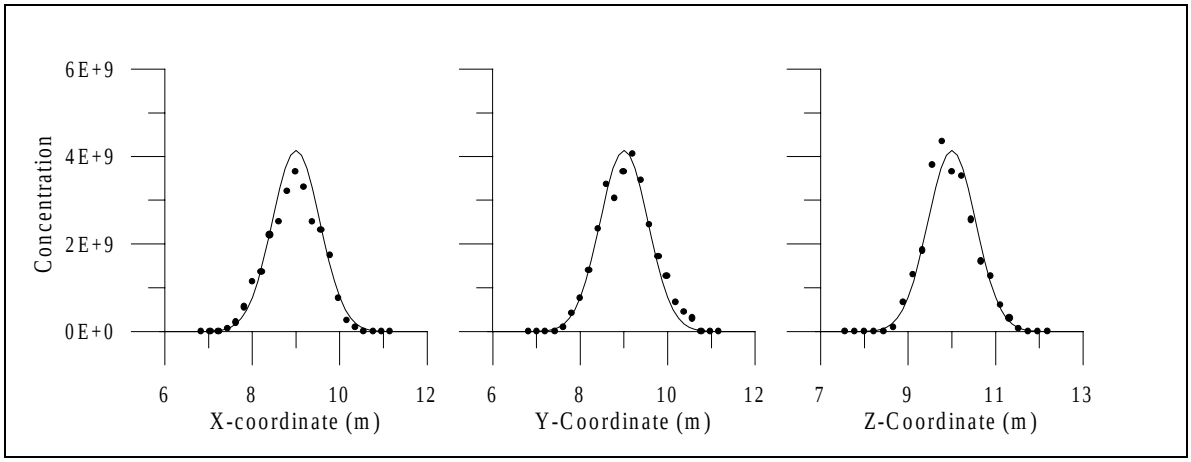

Figure 5.33: Concentration distribution (arbitrary units) resulting from a set of 5,000 particles, using the BC method.

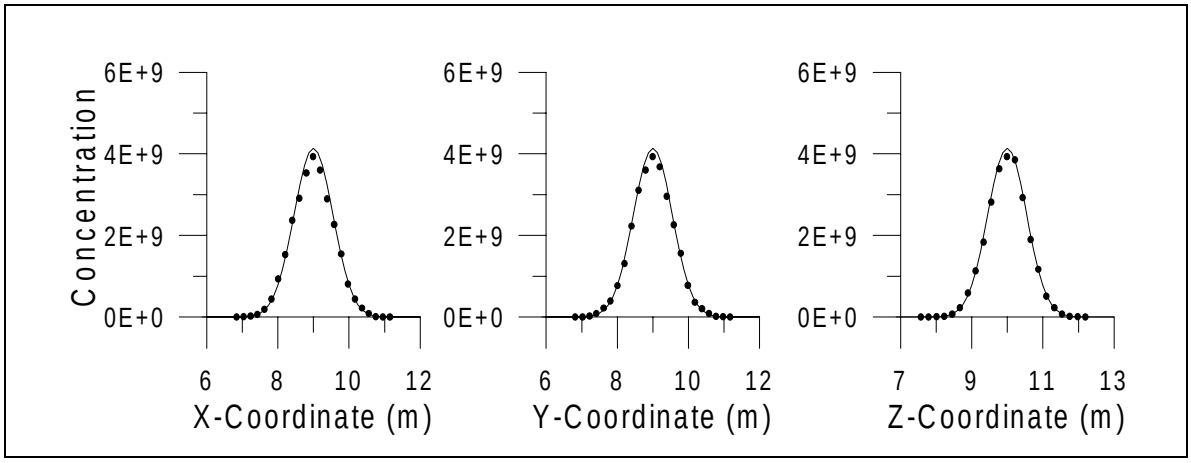

Figure 5.34: Concentration distribution resulting from a set of 50,000 particles, using the BC method.

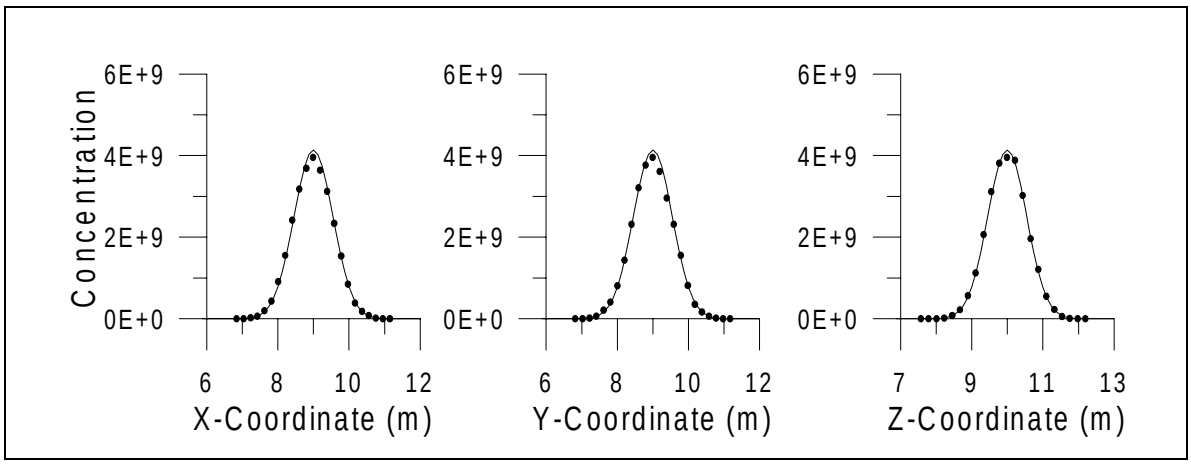

Figure 5.35: Concentration distribution (arbitrary units) resulting from a set of 90,000 particles, using the BC method.

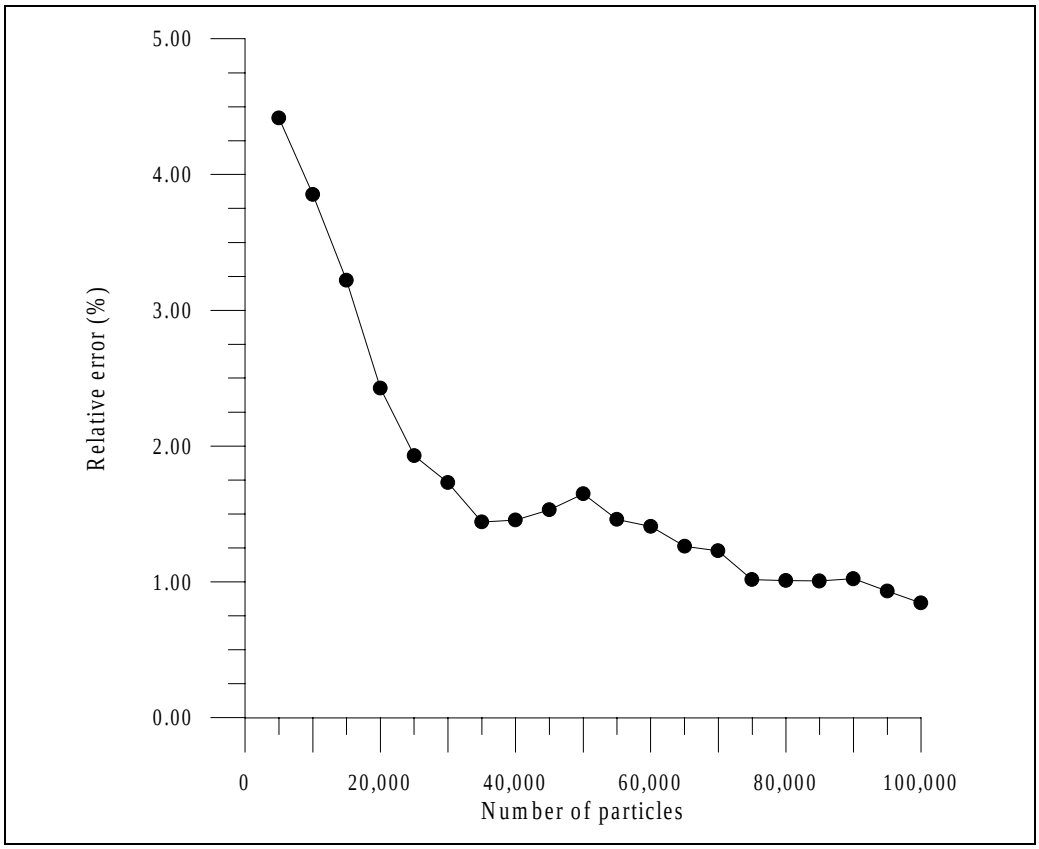

Figure 5.36: Relative error as a function of the number of particles, using the BC transformation method.

Figure 5.37 shows a cross-section of the spherical concentration distribution through the centre, on the horizontal plane, together with lines of equal concentration, for the analytical solution and the modelled data:

Similar analyses have been performed on the same set of 100,000 particles using the kBC and the SPH algorithms.

For the former, introduced mainly to avoid jagged profiles, it was seen that there was a considerable “flattening” of the overall concentration distribution, resulting in a severe underestimation of the values near the centre of the spot. This effect was somehow expected, since it is the consequence of the “flattening” of individual particle mass distributions implied by the algorithm.

The comparison between the analytical Gaussian distribution and that computed by the numerical model using different mass spreading fractions N_1 / N_2 is shown in figures 5.38, 5.39 and 5.40, for various values of l_{Ξ} . Part of the mass located on the principal axes is projected off them due to a non-zero N_2, and the concentration values near the centre become smaller, although the total pollutant mass is conserved.

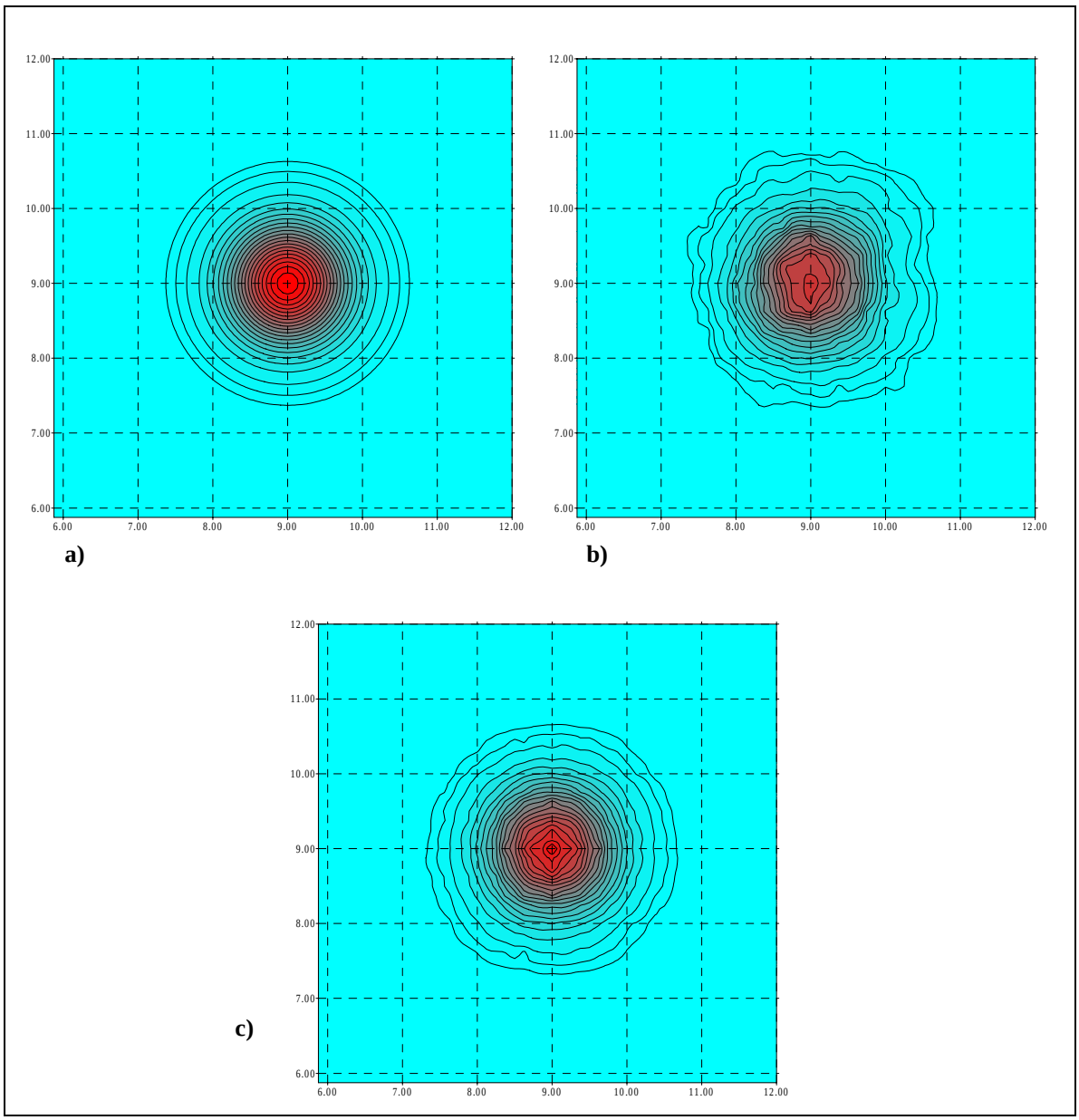

Figure 5.37: Horizontal slice, cutting through the origin, of the concentration distribution resulting from pure diffusion. **a)** analytical solution, **b)** modelled using 10,000 particles, and **c)** modelled using 100,000 particles.

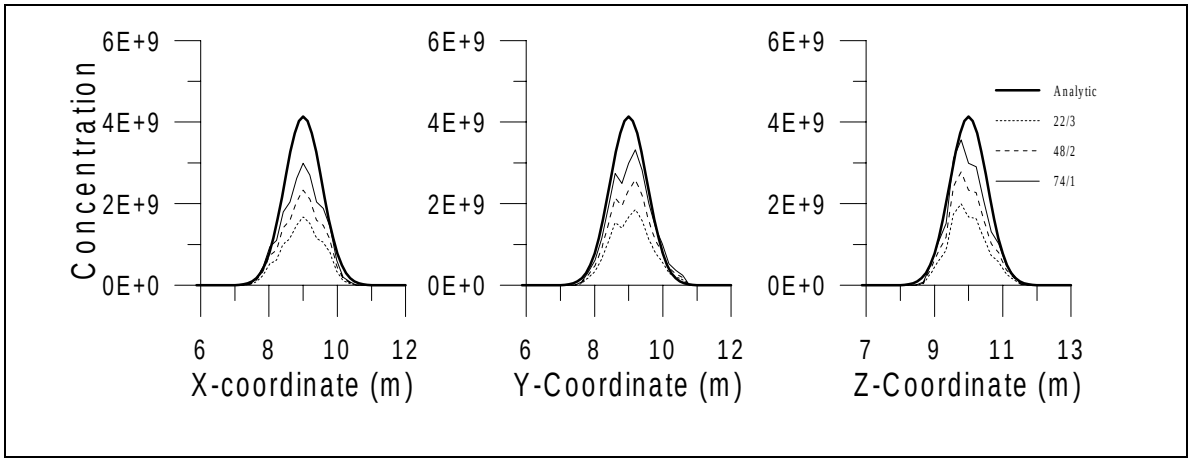

Figure 5.38: Concentration distribution for a set of 100,000 particles, using different ratios N_1 / N_2 for the kBC method, and an integration lengthscale l_Ξ = 0.05m.

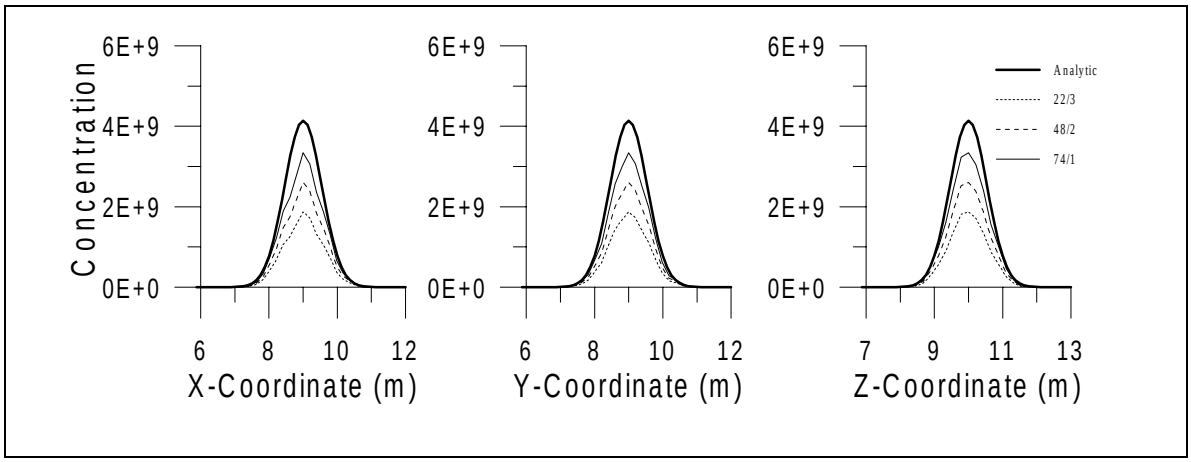

Figure 5.39: Concentration distribution for a set of 100,000 particles, using different ratios N_1 / N_2 for the kBC method, and an integration lengthscale l_Ξ = 0.35m.

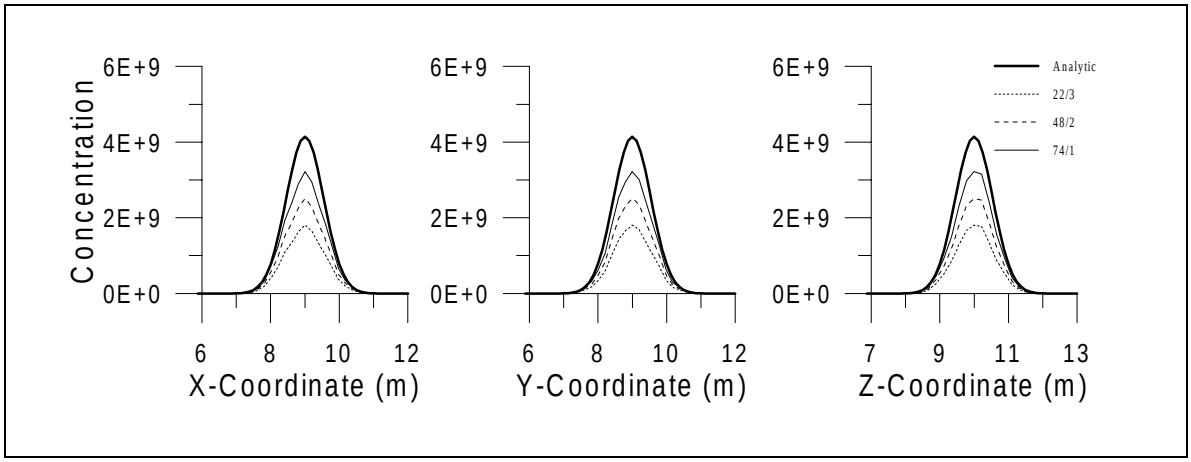

Figure 5.40: Concentration distribution for a set of 100,000 particles, using different ratios N_1/N_2 for the kBC method, and an integration lengthscale l_Ξ = 0.50m.

The effect of the particle spreading on the relative error, as a function of the number of considered particles, is shown in figure 5.41. It can be readily seen that ε_t is larger than that obtained for the BC (N_1 = 100, N_2 = 0) case, and that it becomes less particle-sensitive as a result of the spreading.

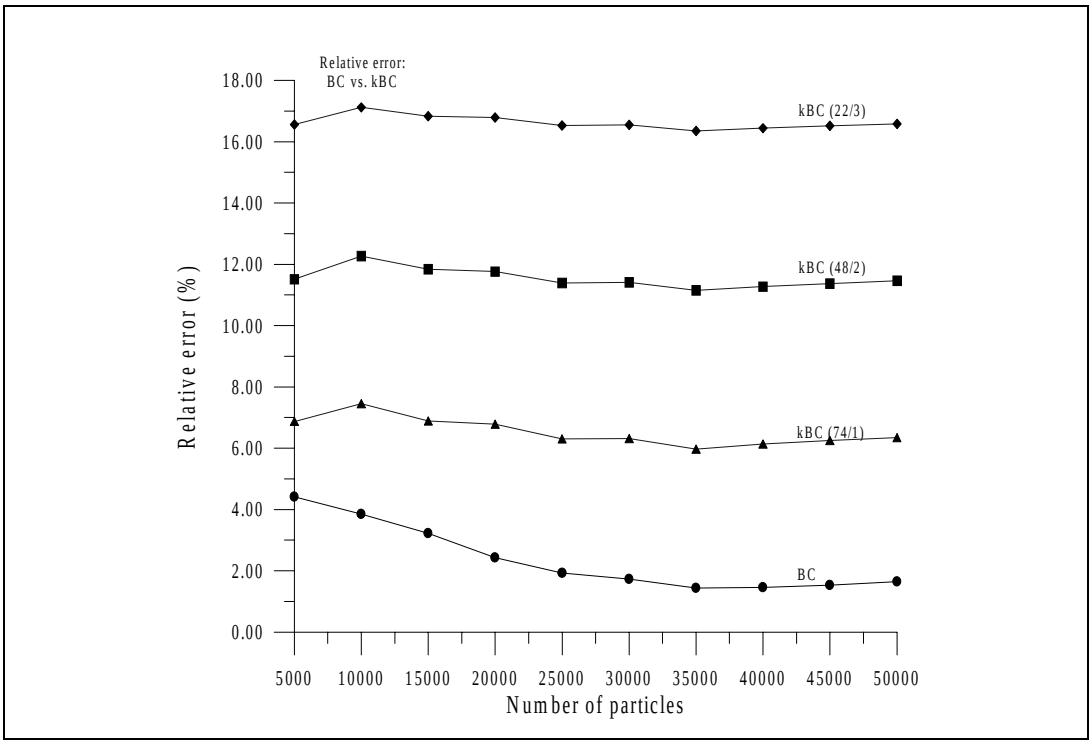

Figure 5.41: Relative error for the kBC method, for different ratios N_1 / N_2., compared to that obtained using the BC method.

Tests with 5,000 and 20,000 particles were done using the Smoothed Particle Hydrodynamics algorithm, in order to determine its sensitivity to the integration lengthscales. The smoothing function was defined as (Gingold and Monaghan, 1982):

$$W_H(r) = \begin{cases} \left(\frac{15}{16l}\right)\left(1-\frac{r^2}{l^2}\right) & r \le l \\ 0 & r \ge l \end{cases} \qquad (5.101)$$

and the integration length l was assumed equal for all particles. Several runs were made to set the range of l for which ε_τ was minimum, and the results are shown in figures 5.42, 5.43, and 5.44, for 5,000 particles, and 5.45, 5.46 and 5.47 for the concentration distributions computed from a 20,000-particle set.

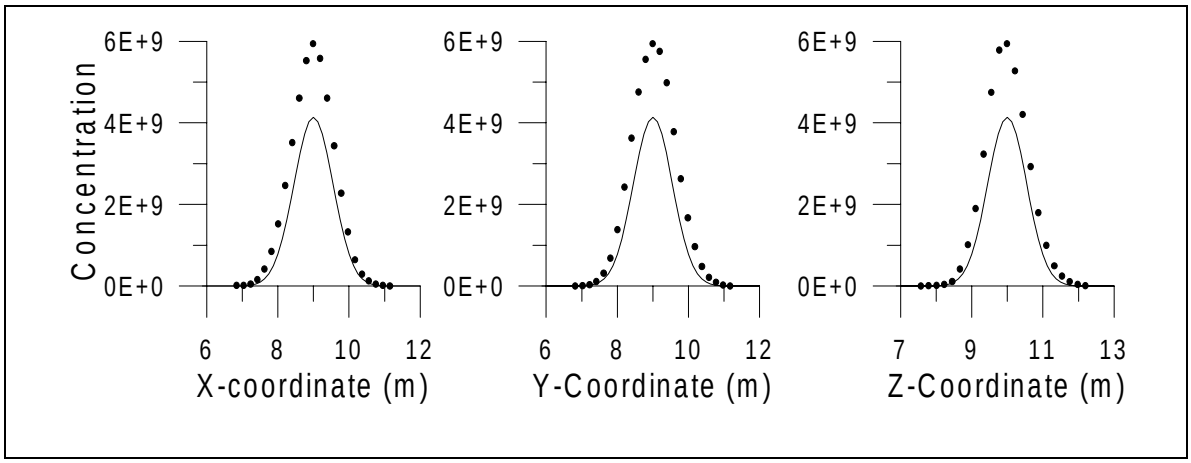

Figure 5.42: Concentration distribution (arbitrary units), calculated for a set of 5,000 particles, using SPH method and an integration length l = 0.60m.

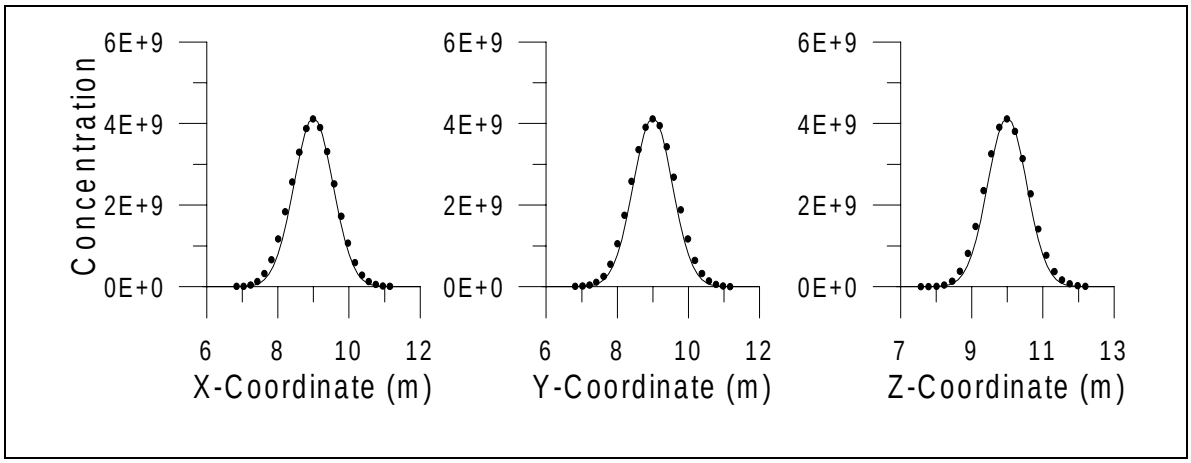

Figure 5.43: Concentration distribution (arbitrary units), calculated for a set of 5,000 particles, using SPH method and an integration length l = 0.70m.

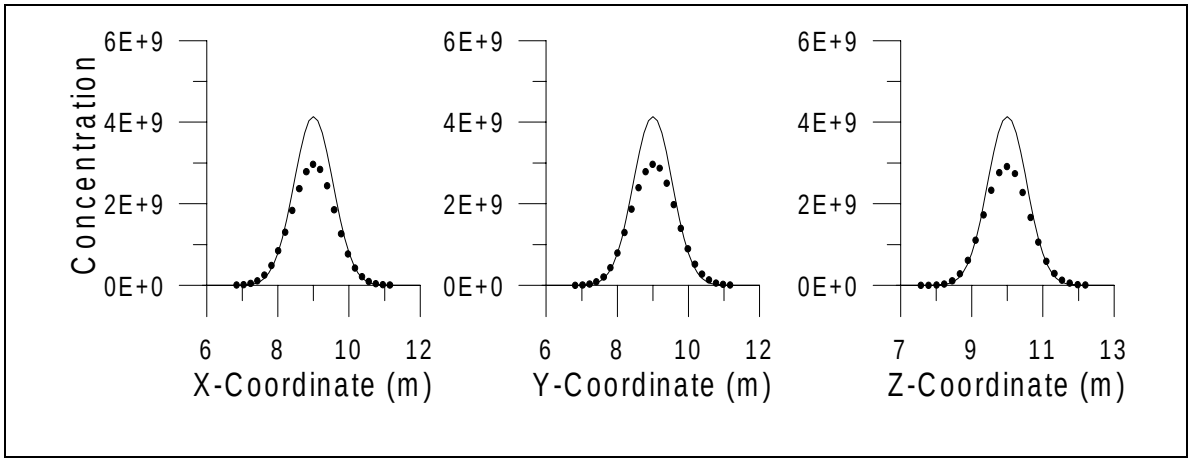

Figure 5.44: Concentration distribution (arbitrary units), calculated for a set of 5,000 particles, using SPH method and an integration length l = 0.80m.

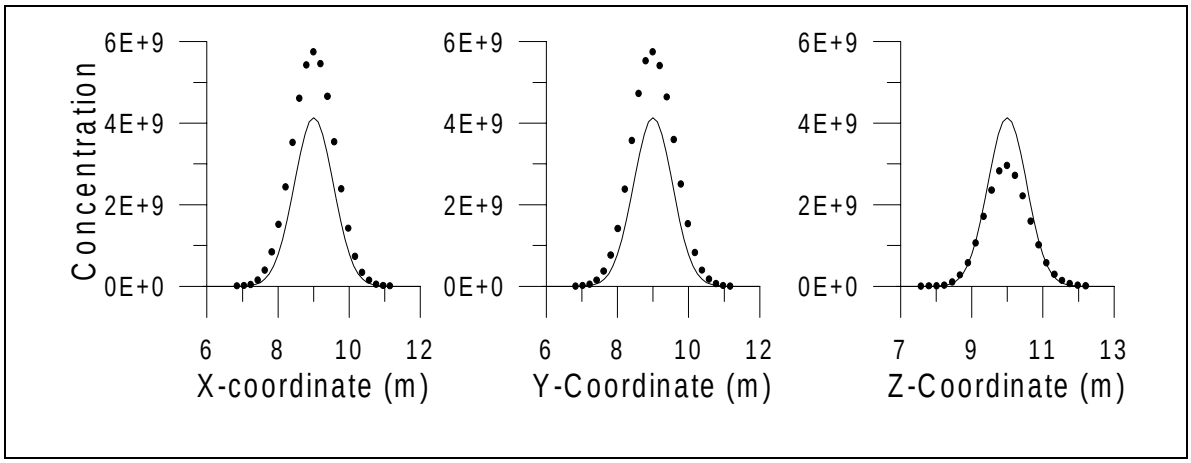

Figure 5.45: Concentration distribution (arbitrary units), calculated for a set of 20,000 particles, using SPH method and an integration length l = 0.60m.

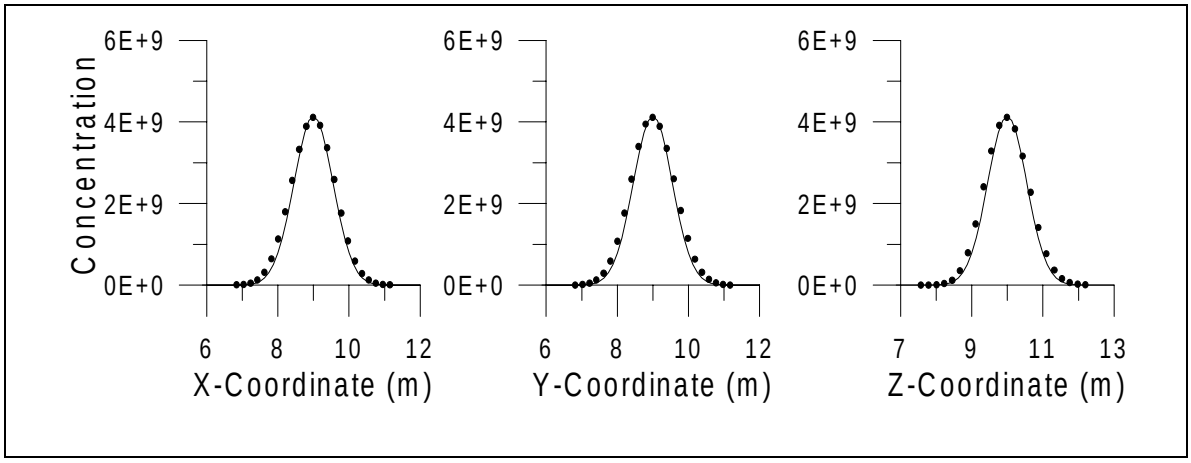

Figure 5.46: Concentration distribution (arbitrary units), calculated for a set of 20,000 particles, using SPH method and an integration length l = 0.70m.

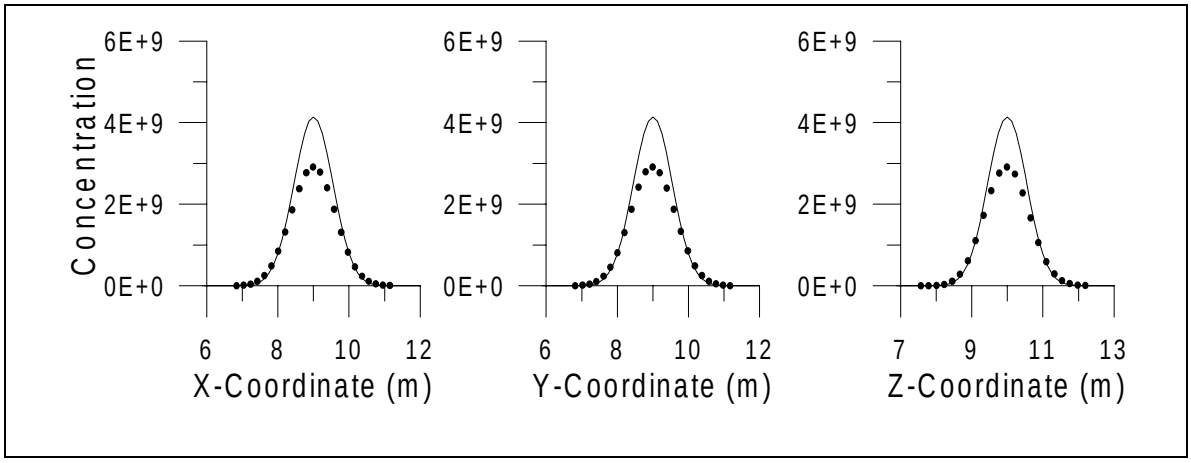

Figure 5.47: Concentration distribution (arbitrary units), calculated for a set of 20,000 particles, using SPH method and an integration length l = 0.80m.

The relative error found in each model run is plotted in figure 5.48 as a function of integration length, both for N = 5,000 and N = 20,000. It is seen that the overall behaviour is similar to that found for the BC algorithm (figure 5.32), and apparently not very dependent on the number of particles in the simulation. It is evident, however, that given a good choice of the integration lengthscale, the BC method requires approximately four times more particles (20,000) than the SPH method (5,000) to achieve a similar error. It appears, therefore, that the SPH method is preferable to the BC algorithm for mapping particle distributions because, by requiring a smaller number of particles, it also demands less computer storage memory and less CPU-time in the transport model run. The higher complexity of the SPH algorithm, specially in the presence of solid boundaries or when individual integration lengthscales are required for each particle, and the need to pre-define a kernel function that is adequate for all cases of interest, renders this method somewhat awkward to use in situations where a highly accurate results are not needed.

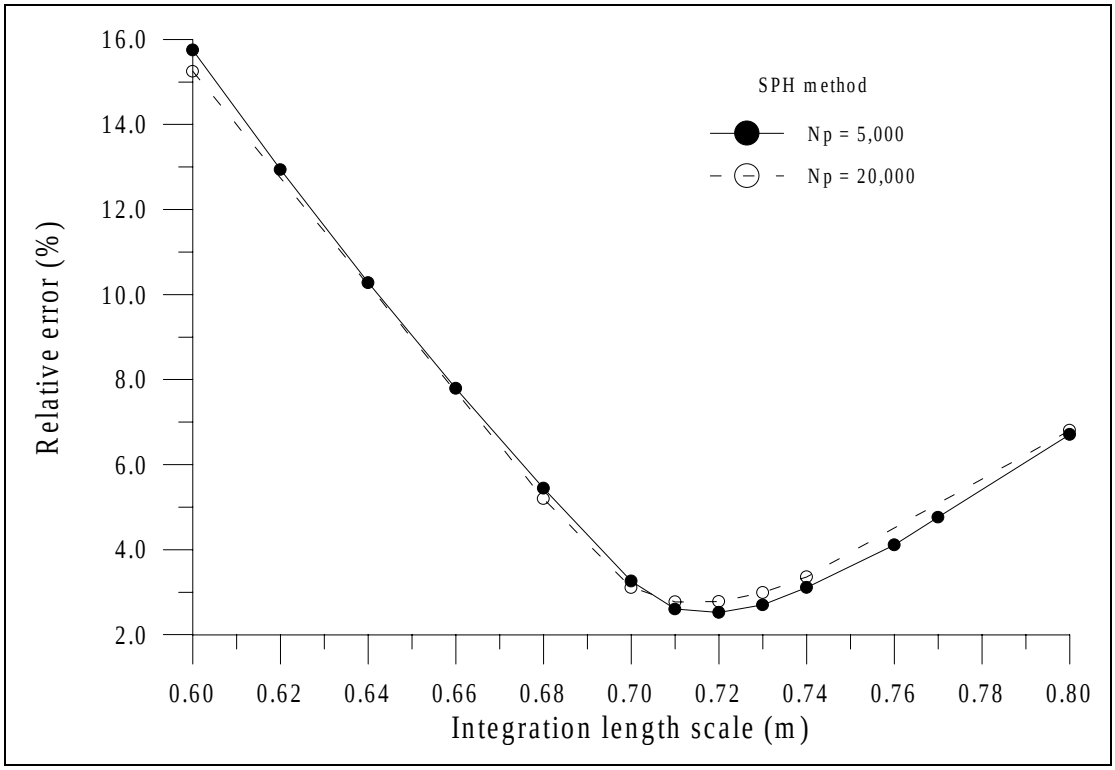

Figure 5.48: Relative error as a function of the integration lengthscale, for a set of 5,000 particles, and one of 20,000 particles.

Figures 5.49, 5.50 and 5.51 show the concentration profiles obtained using the kernel function defined in (5.101), an integration length l = 0.72m, and different numbers of particles. From these, and the previous figures, it seems that the results given by the kernel method are more l-dependent than N-dependent, as occurred for the BC method when N was larger than a certain value. This suggests that a (much smaller) threshold value for N also exists in the SPH algorithm.

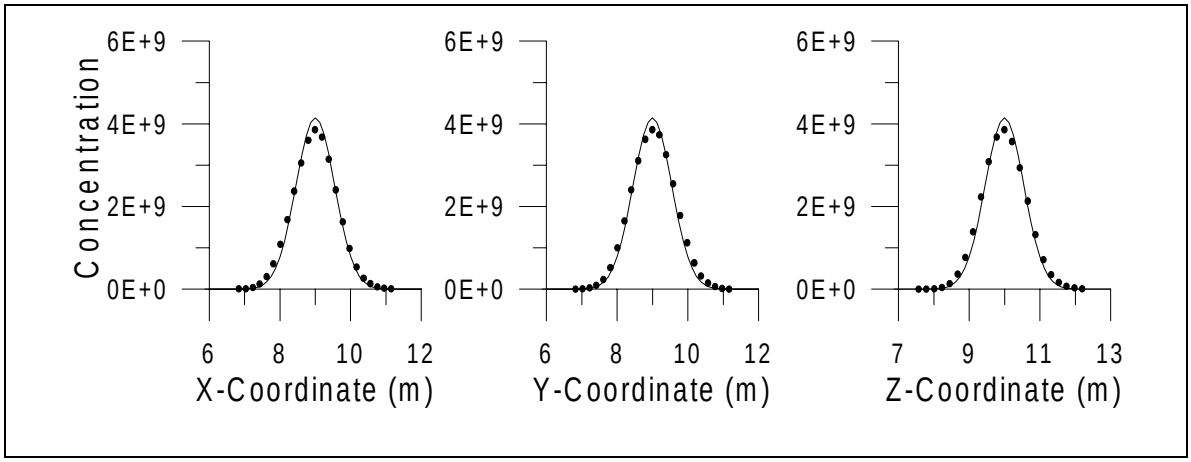

Figure 5.49: Concentration distribution (arbitrary units), calculated using SPH method and an integration length l = 0.72m, for a set of 7,000 particles.

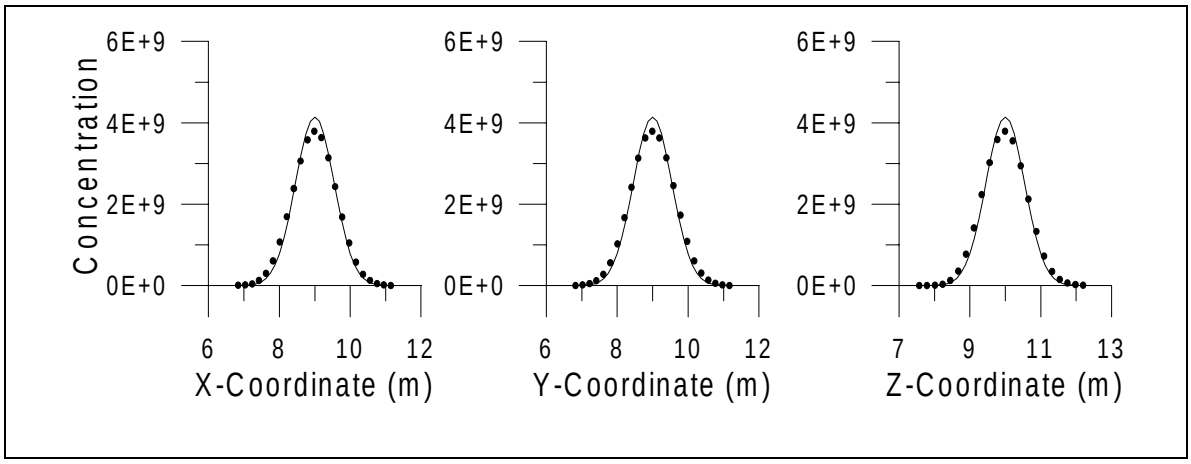

Figure 5.50: Concentration distribution (arbitrary units), calculated using SPH method and an integration length l = 0.72m, for a set of 15,000 particles.

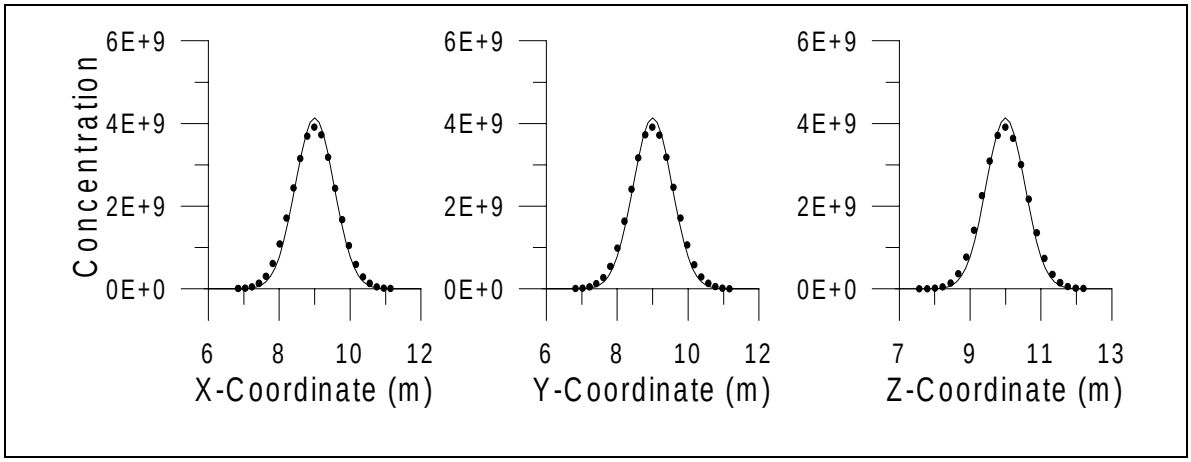

Figure 5.51: Concentration distribution (arbitrary units), calculated using SPH method and an integration length l = 0.72m, for a set of 27,000 particles.

5.3.4 One-dimensional dispersion of a continuous discharge of duration τ:

In a system in which physical transport is effected primarily by a one-dimensional steady and uniform flow field, in which the effects of dispersion are spatially constant, and concentrations can be considered horizontally and vertically well-mixed, the advection-dispersion equation is reduced to

$$\frac{\partial C}{\partial t} = -u\frac{\partial C}{\partial x} + D\frac{\partial^2 C}{\partial x^2} \tag{5.102}$$

For the case of a continuous source of infinite duration, and after imposing the following initial and boundary conditions,

$$C(x,0) = 0 \quad \text{for} \quad x \geq 0 \tag{5.103a}$$

$$C(0,t) = C_0 \quad \text{for} \quad t \geq 0 \tag{5.103b}$$

$$C(\infty,t) = 0 \quad \text{for} \quad t \geq 0 \tag{5.103c}$$

Ogata and Banks (1961) derived an analytical solution for conservative substances in the form

$$C(x,t) = \frac{C_0}{2}\left[\operatorname{erfc}\left(\frac{x-ut}{2\sqrt{Dt}}\right) + \exp\left(\frac{ux}{D}\right)\operatorname{erfc}\left(\frac{x+ut}{2\sqrt{Dt}}\right)\right] \tag{5.104}$$

where *erfc* stands for the complementary error function, defined by

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt \tag{5.105}$$

When the conservative substance is discharged during a finite period τ, the concentration at any time smaller than τ is given by (Runkel, 1996):

$$C(x,t) = \frac{C_0}{2}\left\{\left[\operatorname{erfc}\left(\frac{x-ut}{2\sqrt{Dt}}\right) - \operatorname{erfc}\left(\frac{x-u(t-\tau)}{2\sqrt{D(t-\tau)}}\right)\right] + \exp\left(\frac{ux}{D}\right)\left[\operatorname{erfc}\left(\frac{x+ut}{2\sqrt{Dt}}\right) - \operatorname{erfc}\left(\frac{x+u(t-\tau)}{2\sqrt{D(t-\tau)}}\right)\right]\right\} \quad \text{for } t<\tau \tag{5.106}$$

after assuming that the following conditions apply:

$$C(x,0) = 0 \quad \text{for} \quad x \geq 0 \tag{5.107a}$$

$$C(0,t) = C_0 \quad \text{for} \quad \tau \geq t \geq 0 \tag{5.107b}$$

$$C(0,t) = 0 \quad \text{for} \quad t > \tau \tag{5.107c}$$

$$C(\infty,t) = 0 \quad \text{for} \quad t \geq 0 \tag{5.107d}$$

For larger values of t, the solution for $C(x,t)$ is again given by equation (5.104).

Two situations were considered for modelling. In the first one, concentrations were calculated as a function of the distance to the source, at a given time t_k, which was chosen to be smaller than τ in some cases, and greater in others; in the second case, the concentration was calculated at a fixed distance x_k from the source, as a function of time. Both the box-counting (BC) method and the kernel (SPH) algorithm were tested.

The following twelve figures correspond to the first situation. The data were obtained by modelling the continuous discharge of a point source in a one-dimensional flow, with a current velocity u = 0.3 m/s, a diffusion coefficient D = 2 m^2/s, 10216 particles, and a timestep Δt = 0.05 seconds, for a total period of 30 seconds, smaller than the duration of the discharge. The concentration distribution in figures 5.52, 5.53 and 5.54 was computed using the BC method, with a 5 m resolution, and an integration length of 2 m, 25 m, and 50 m, respectively. Figures 5.55, 5.56 and 5.57 were obtained with the SPH method, prescribing the same spatial resolution, and with integration lengthscales l equal to 1 m, 7 m, and 15 m, respectively:

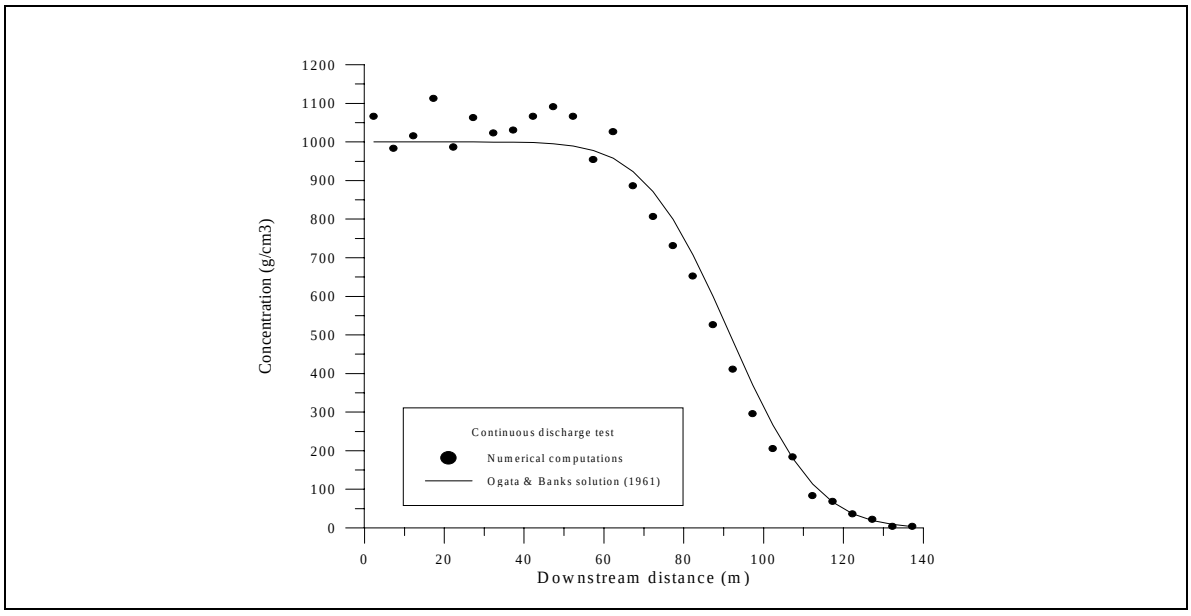

Figure 5.52: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using BC method, with an integration length of 2 m.

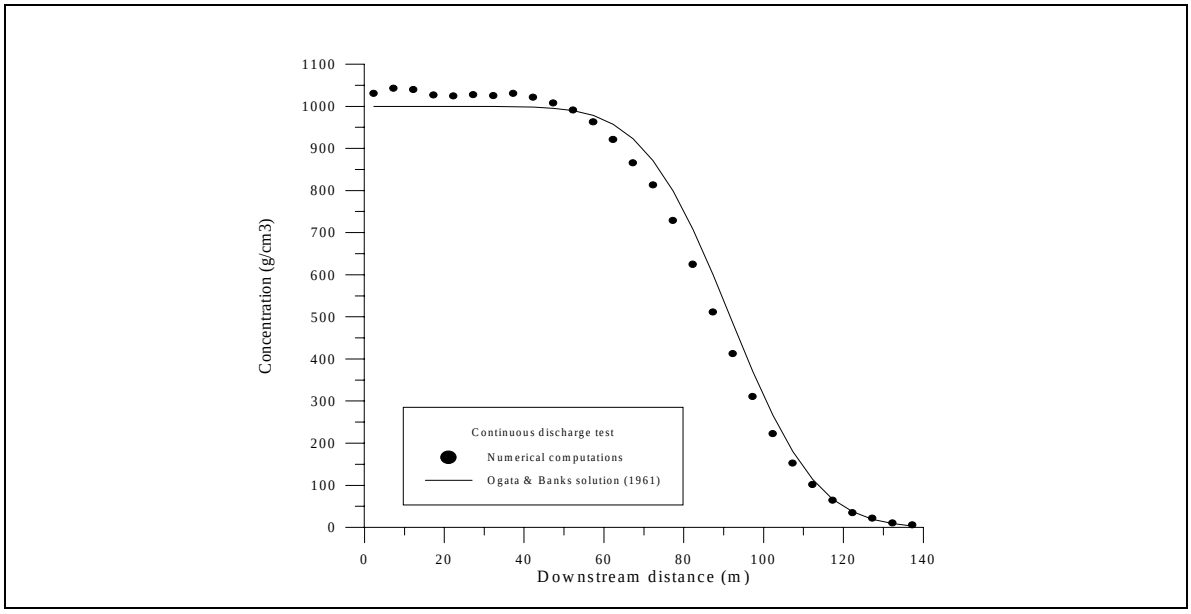

Figure 5.53: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using BC method, with an integration length of 25 m.

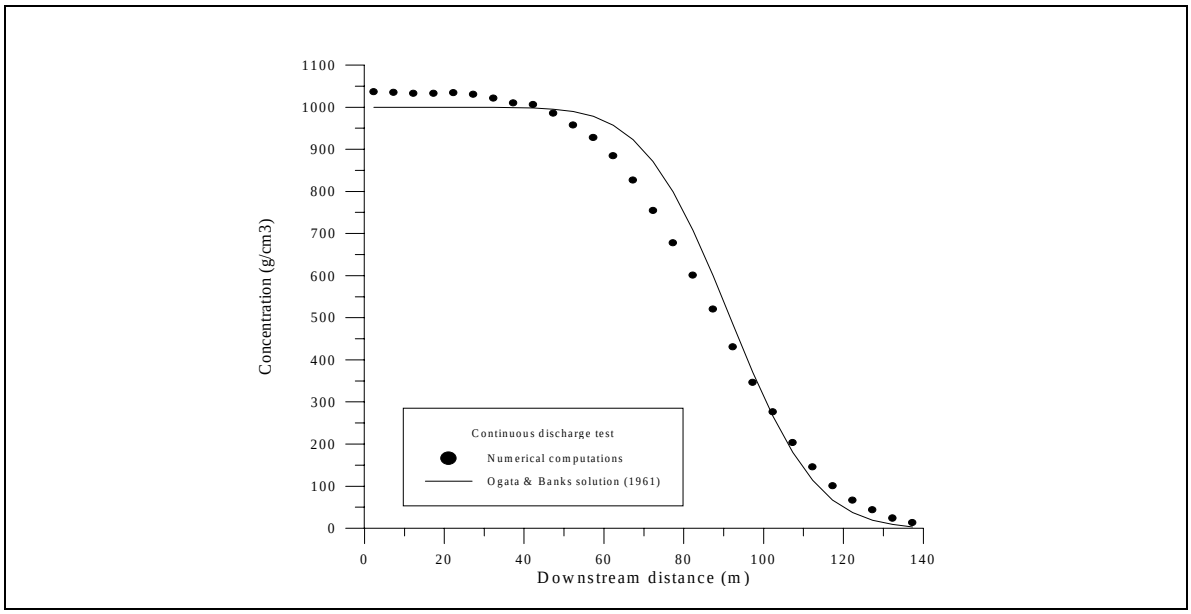

Figure 5.54: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using BC method, with an integration length of 50 m.

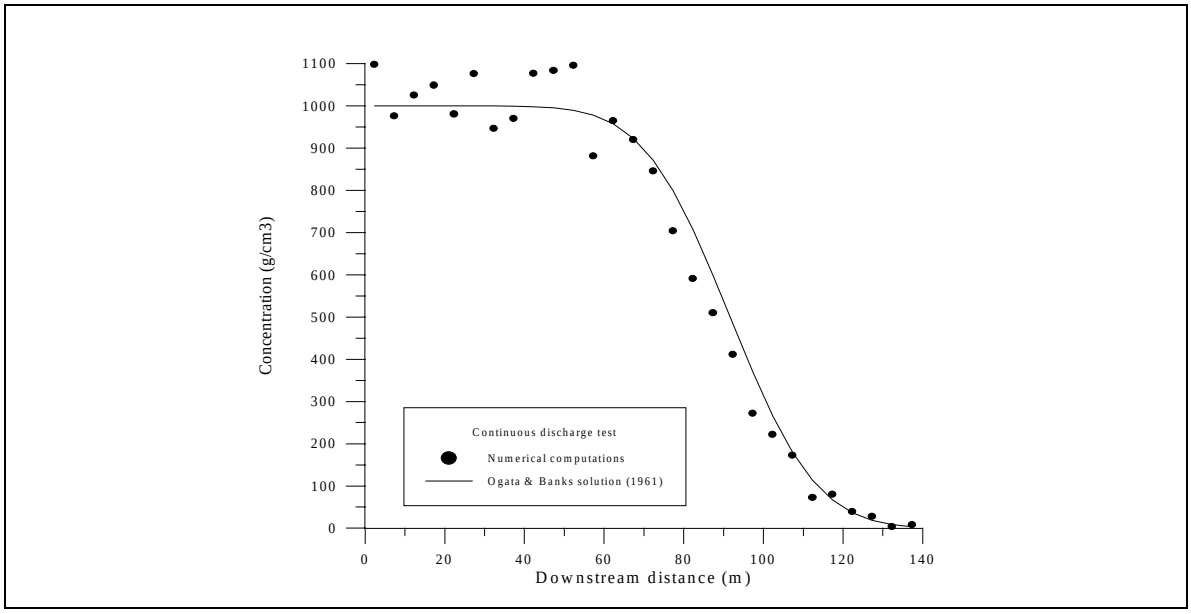

Figure 5.55: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using SPH method, with an integration length of 1 m.

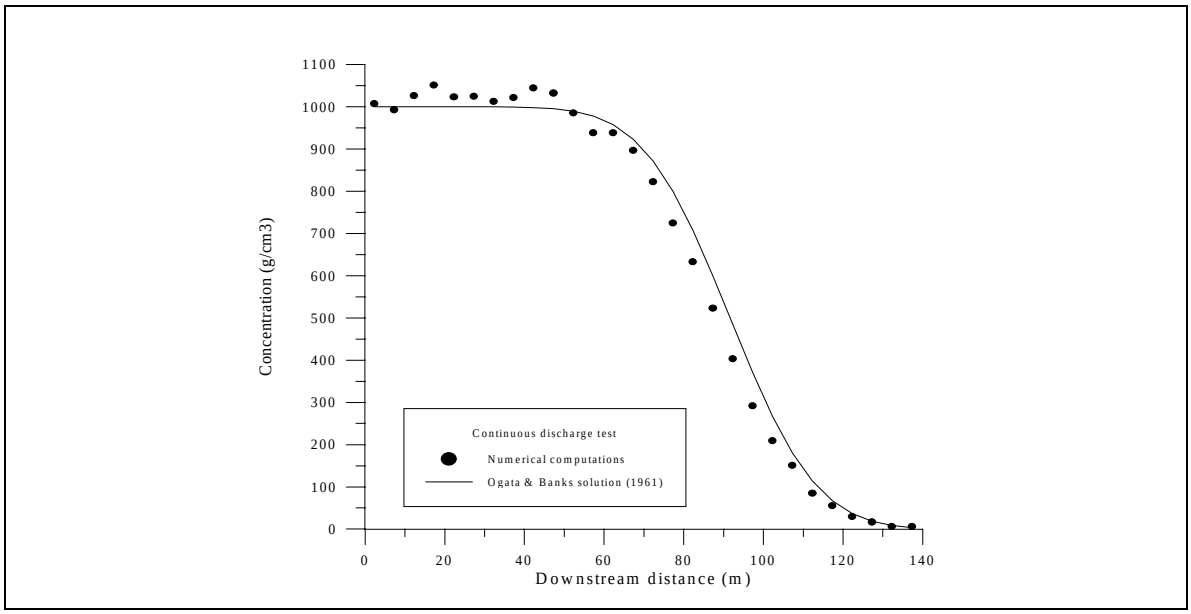

Figure 5.56: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using SPH method, with an integration length of 7 m.

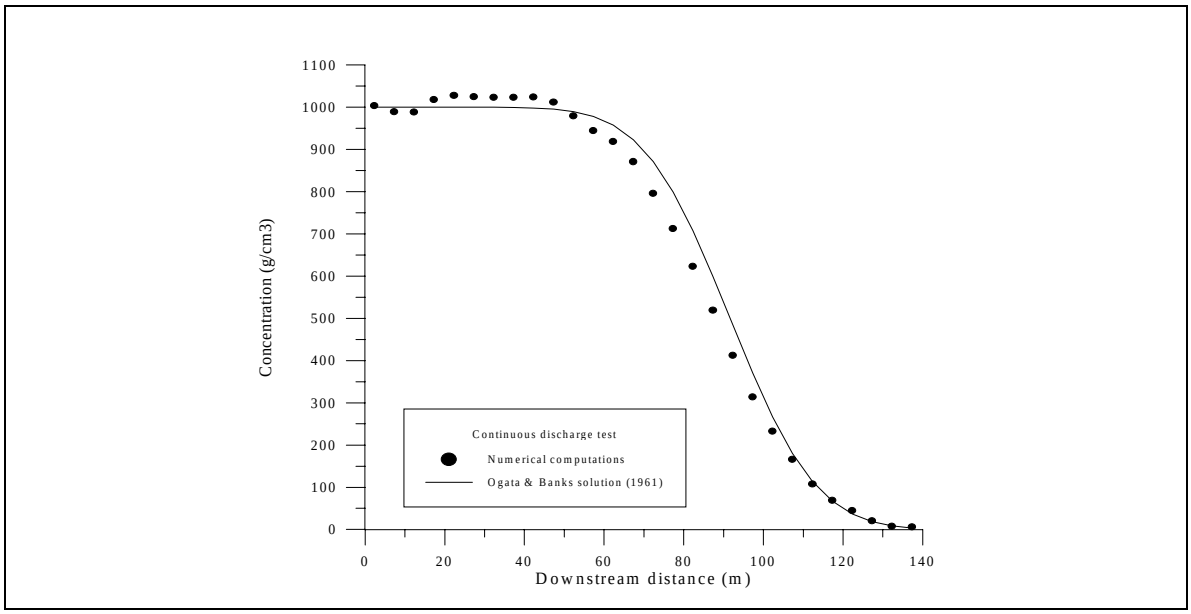

Figure 5.57: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, using SPH method, with an integration length of 15 m.

The error in the concentration distribution, as a function of the integration lengthscale, is given in figure 5.58. Although the general trend is as expected, with a lengthscale for which the error is minimum, the behaviour of the error found from the SPH algorithm is somewhat different, since the values of ε are very similar for a rather wide range of *l*. For this particular case, then, the concentration computation using SPH is not as dependent on the integration length as it is using BC, although the results are not as accurate, as revealed by the slightly larger error.

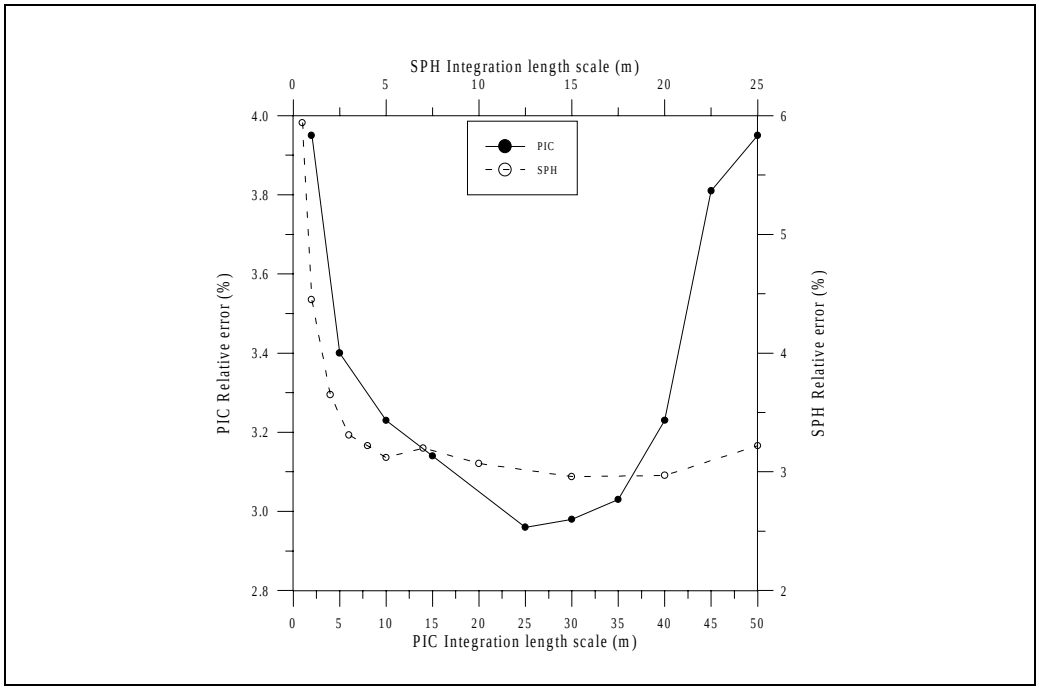

Figure 5.58: Relative error as a function of integration lengthscale, for BC and SPH methods.

The computed concentration distribution for the case of $t > \tau$ (i.e., when the time at which the concentration is evaluated is larger than the duration of the discharge) are shown in figures 5.59 to 5.64, in which the continuous curve represents the analytical solution. The physical parameters of the model run are equal to those in the previous case; the resolution of the computed distribution is 5 m, and the integration length is 2 m, 15 m, and 40 m, respectively for the first three figures, obtained with the BC method, and 2 m, 8 m, and 15 m for the last three plots, using the SPH mapping method.

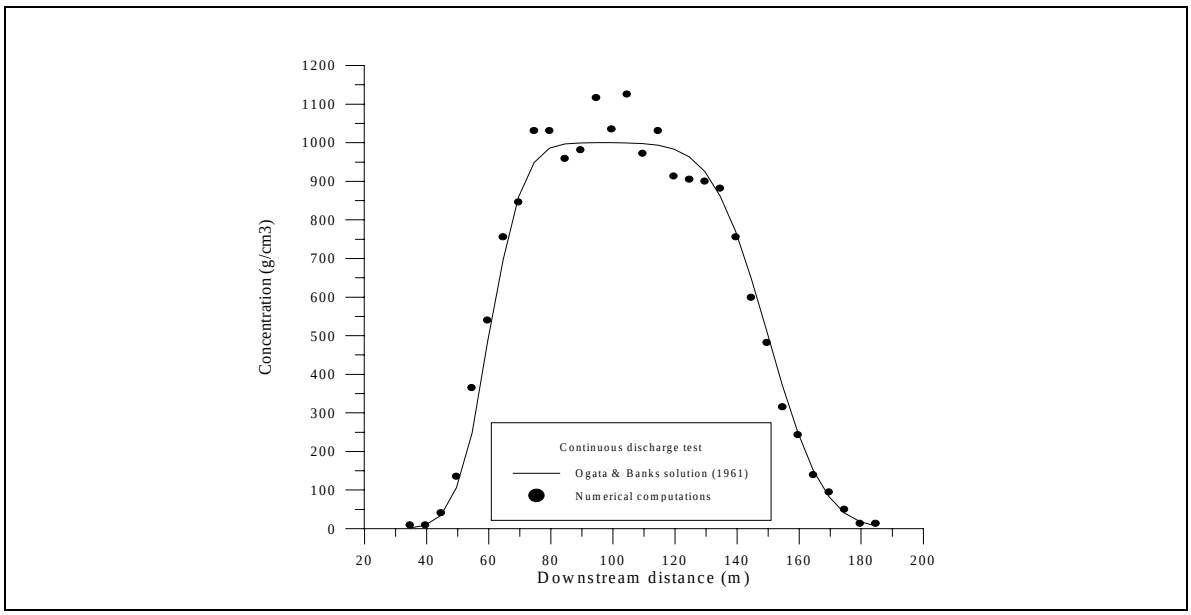

Figure 5.59: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using BC method, with an integration length of 2 m.

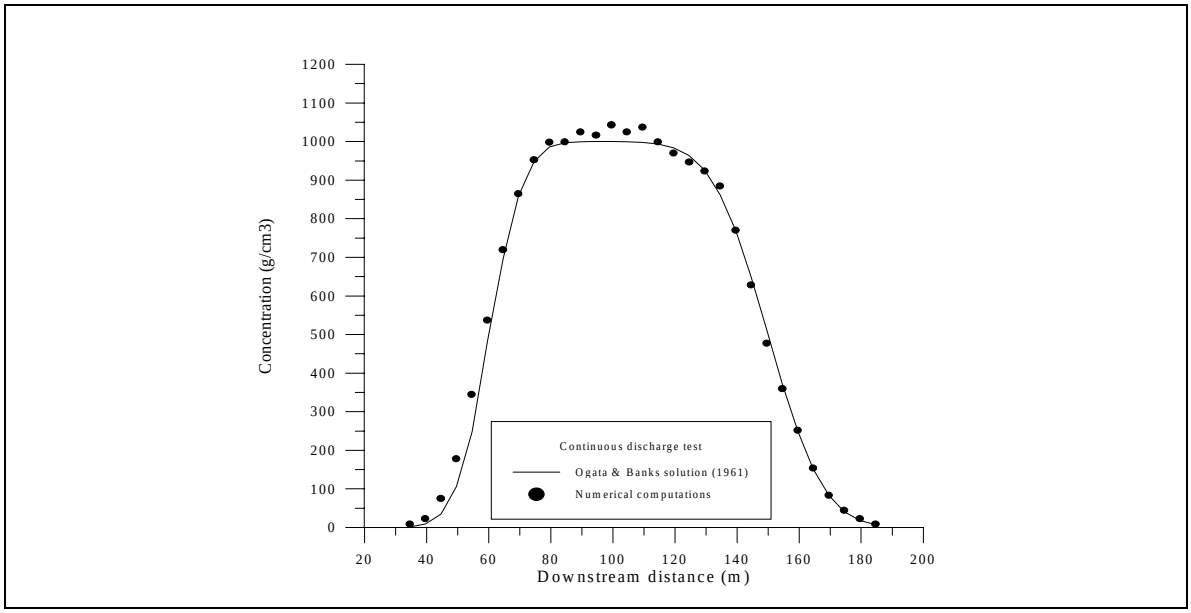

Figure 5.60: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using BC method, with an integration length of 15 m.

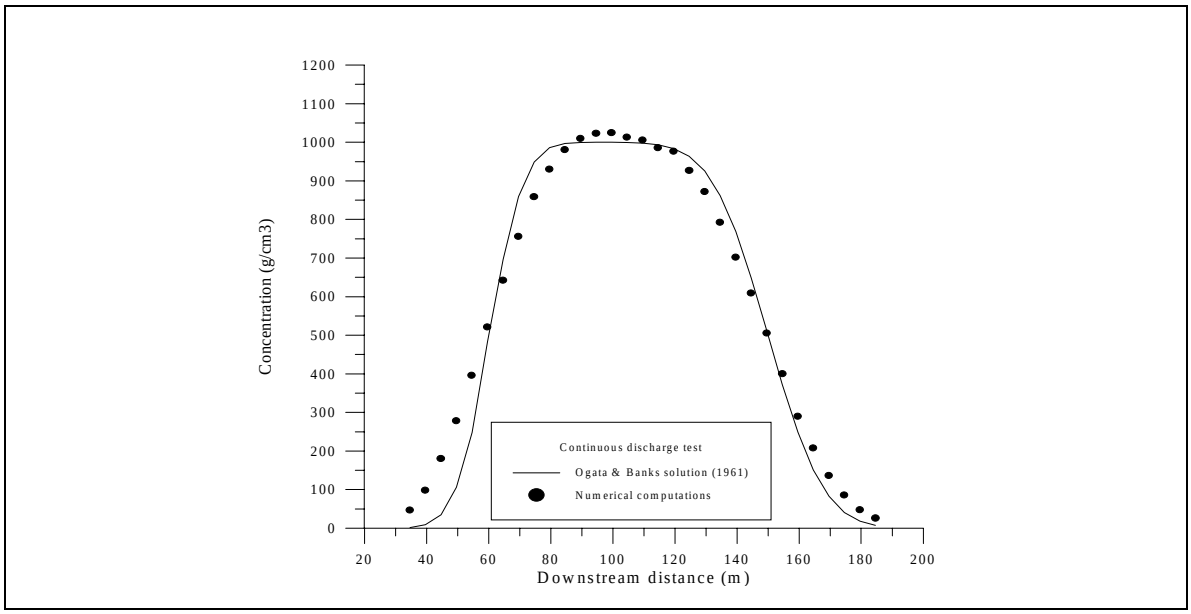

Figure 5.61: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using BC method, with an integration length of 40 m.

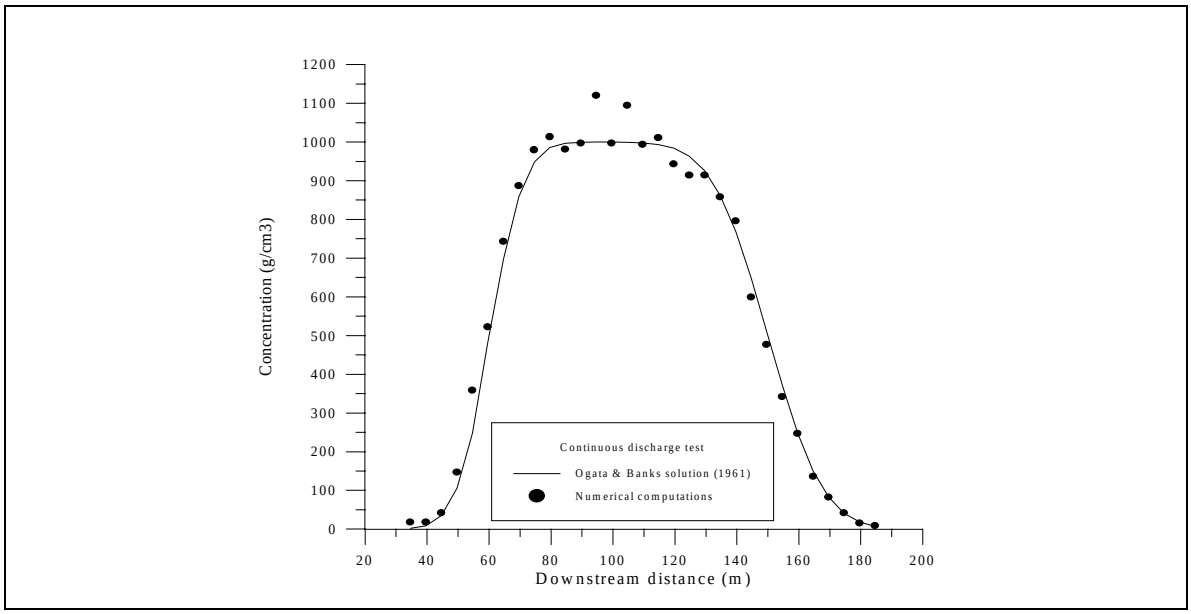

Figure 5.62: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using SPH method, with an integration length of 2 m.

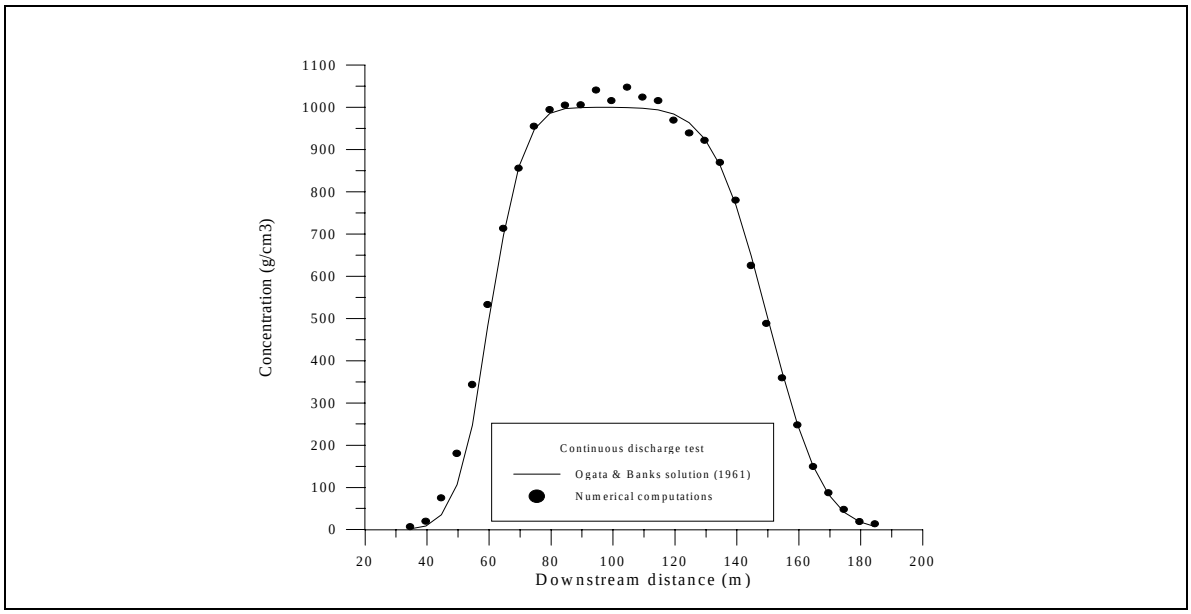

Figure 5.63: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using SPH method, with an integration length of 8 m.

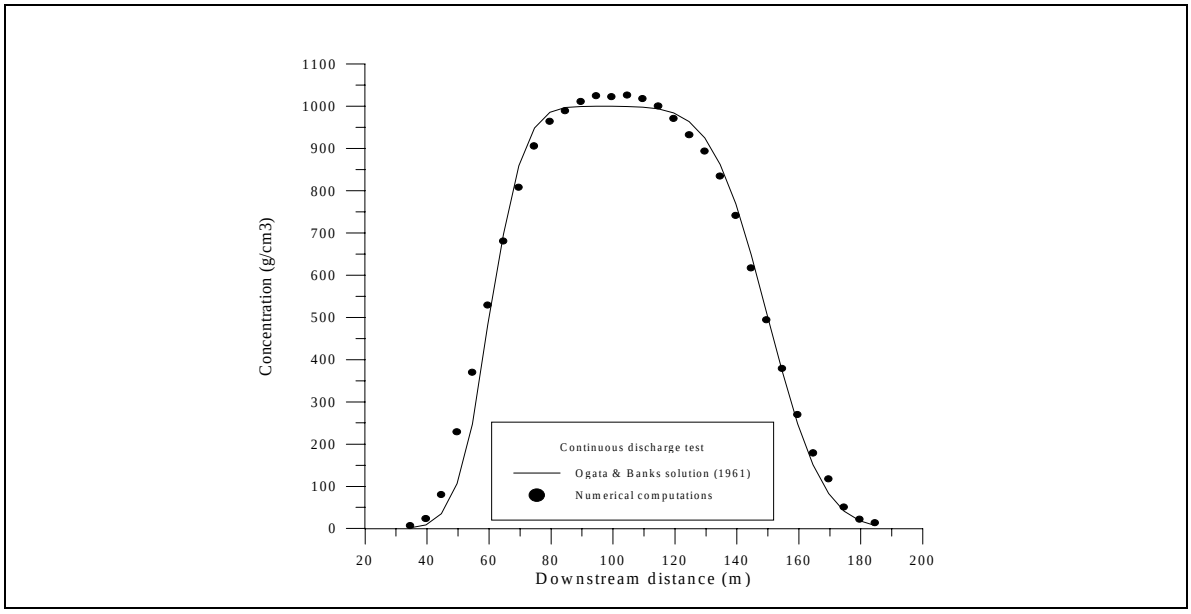

Figure 5.64: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, when $t > \tau$, using SPH method, with an integration length of 15 m.

The relative error obtained using both mapping methods is shown in figure 5.65. Unlike the previous case, here the behaviour of both functions is very similar, and follows the general pattern found before.

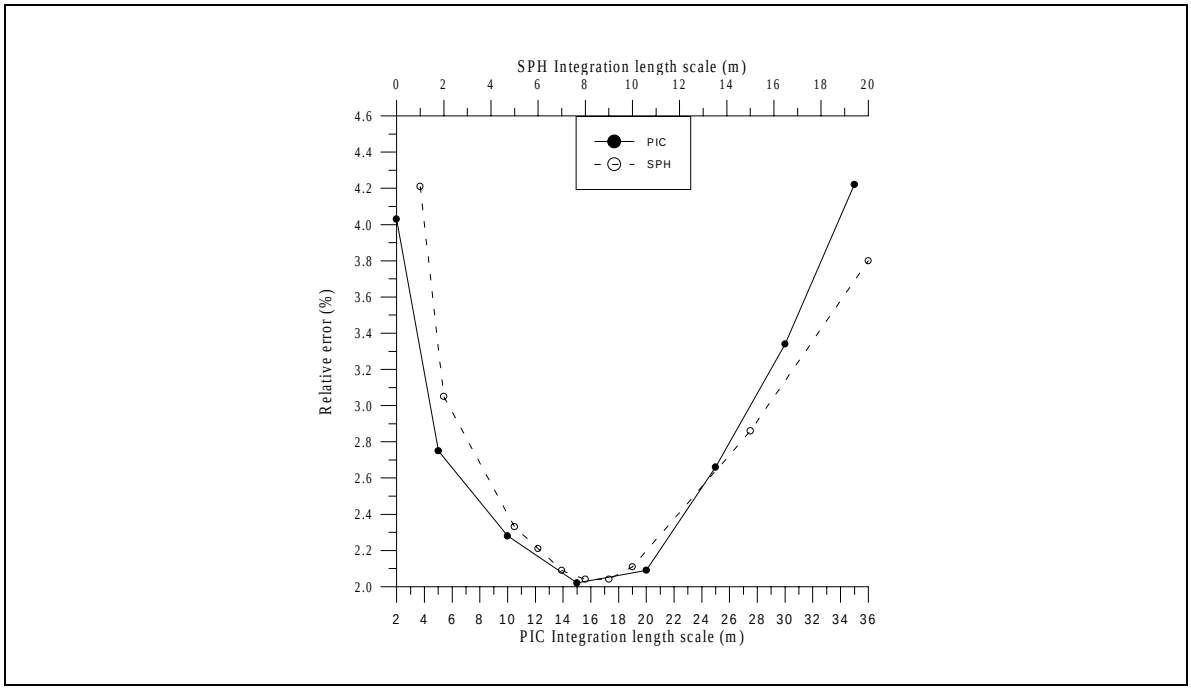

Figure 5.65: Relative error as a function of integration lengthscale, for the BC and the SPH methods.

The second situation, which considers the time-variation of the concentration at a fixed position, as the pollutant cloud is dragged past by the current, is illustrated in figures 5.66 to 5.71. To obtain hese distributions, a total of 8,000 particles were released, in a 3 m/s current, and the diffusion coefficient was taken to be D = 7 m^2/s. Again, the BC method was used to calculate the distribution shown in the first three plots, with integration lengths of 2 m, 10 m, and 20 m, respectively, and the SPH method was employed for the rest, using the same integration lengthscales.

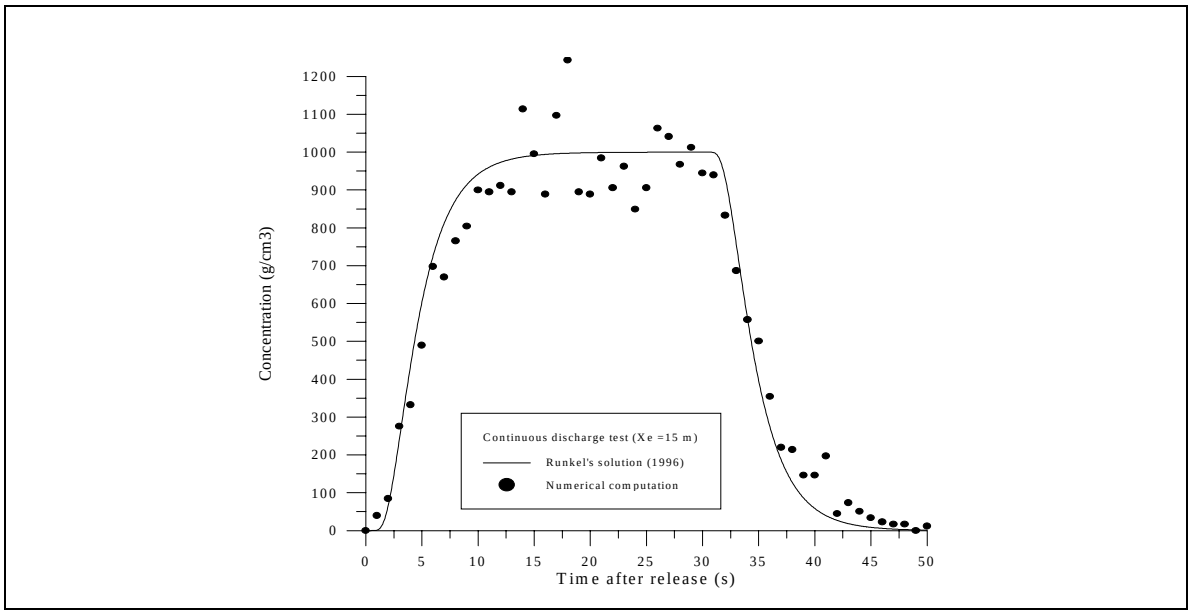

Figure 5.66: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using BC method, with an integration length of 2 m.

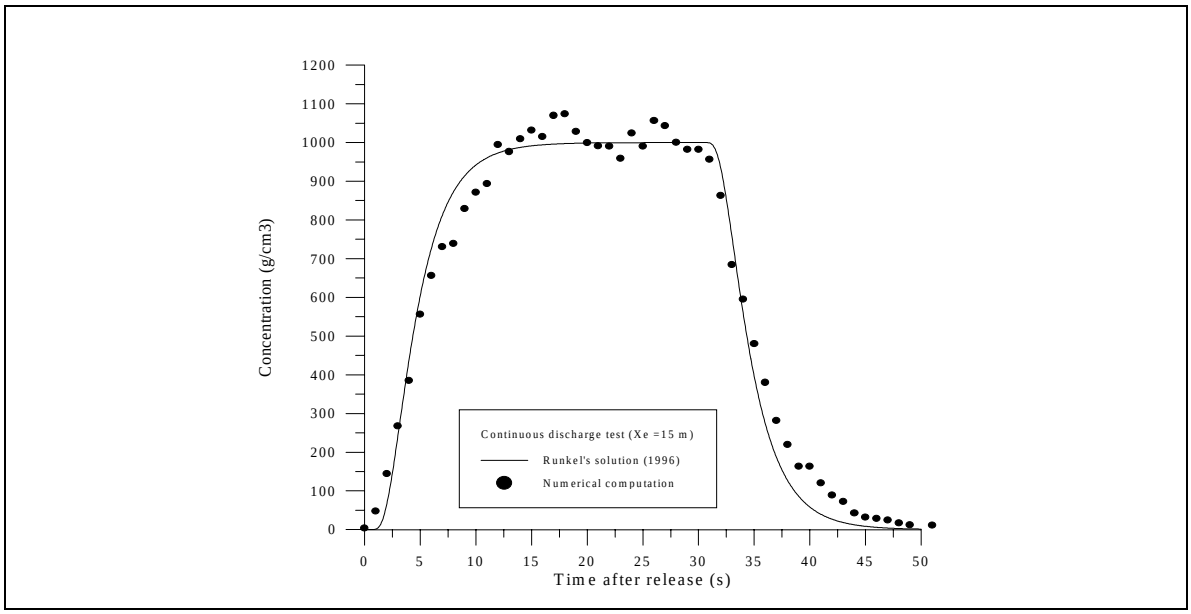

Figure 5.67: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using BC method, with an integration length of 10 m.

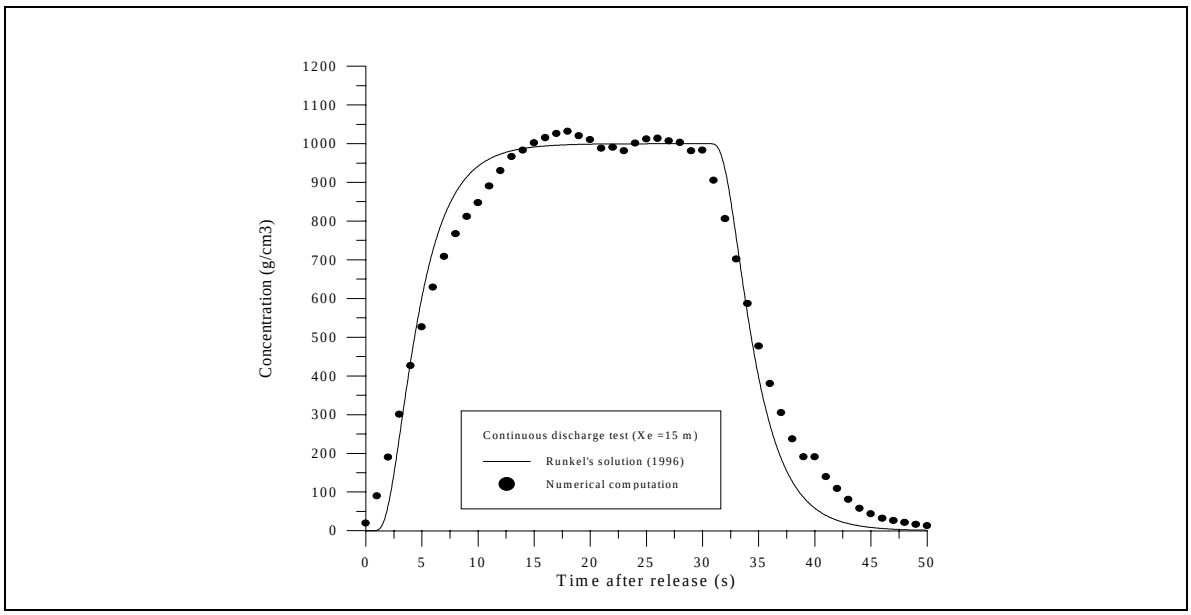

Figure 5.68: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using BC method, with an integration length of 20 m.

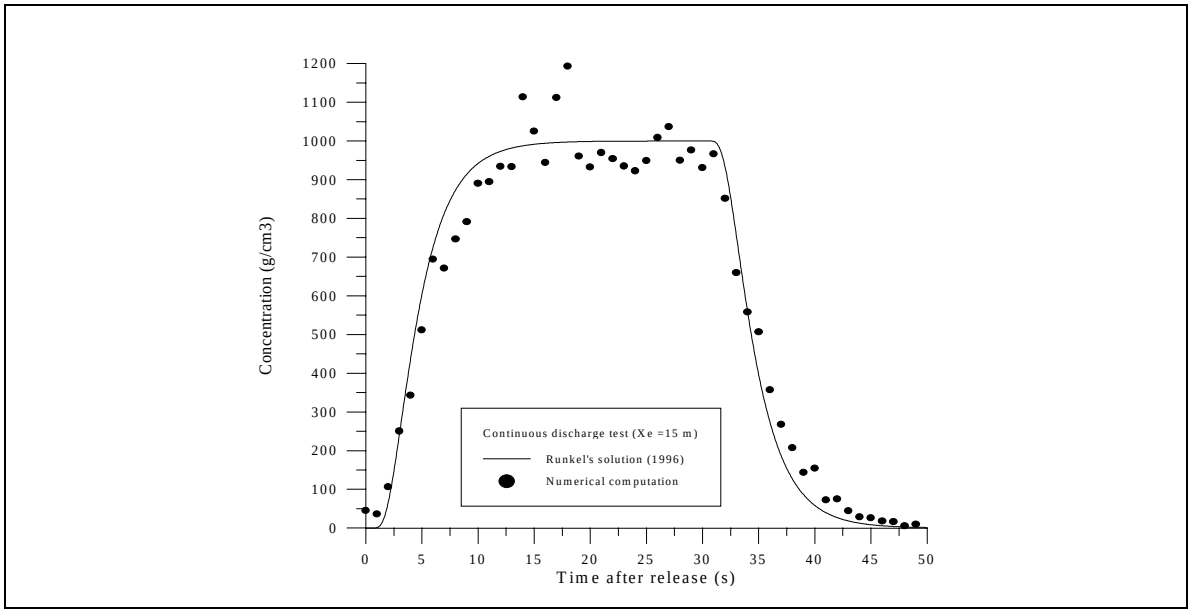

Figure 5.69: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using SPH method, with an integration length of 2 m.

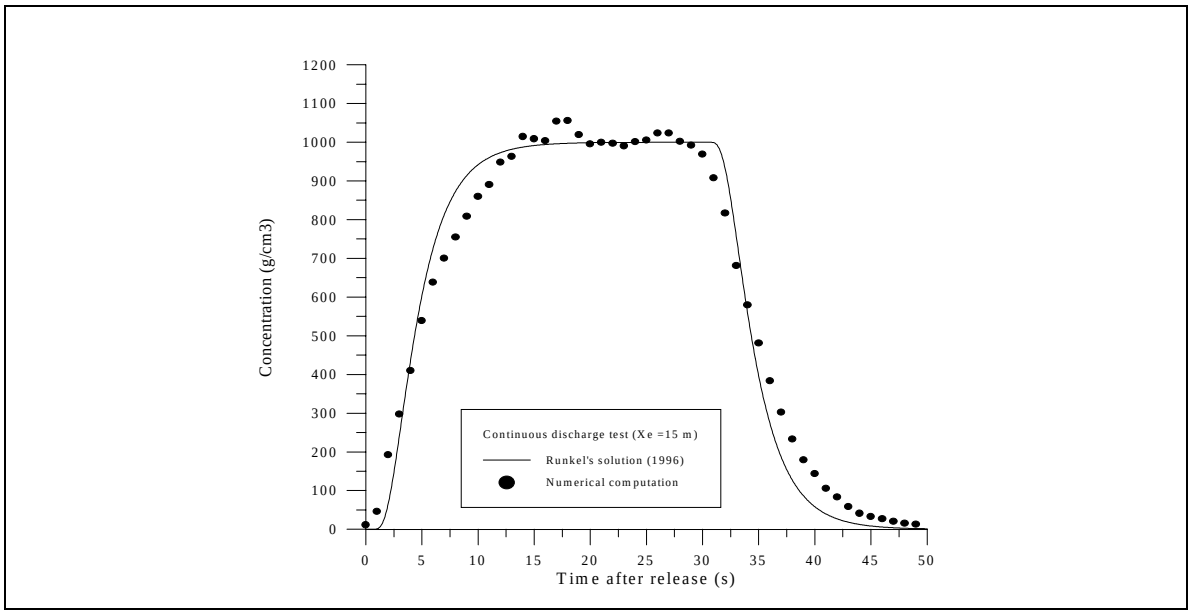

Figure 5.70: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using SPH method, with an integration length of 10 m.

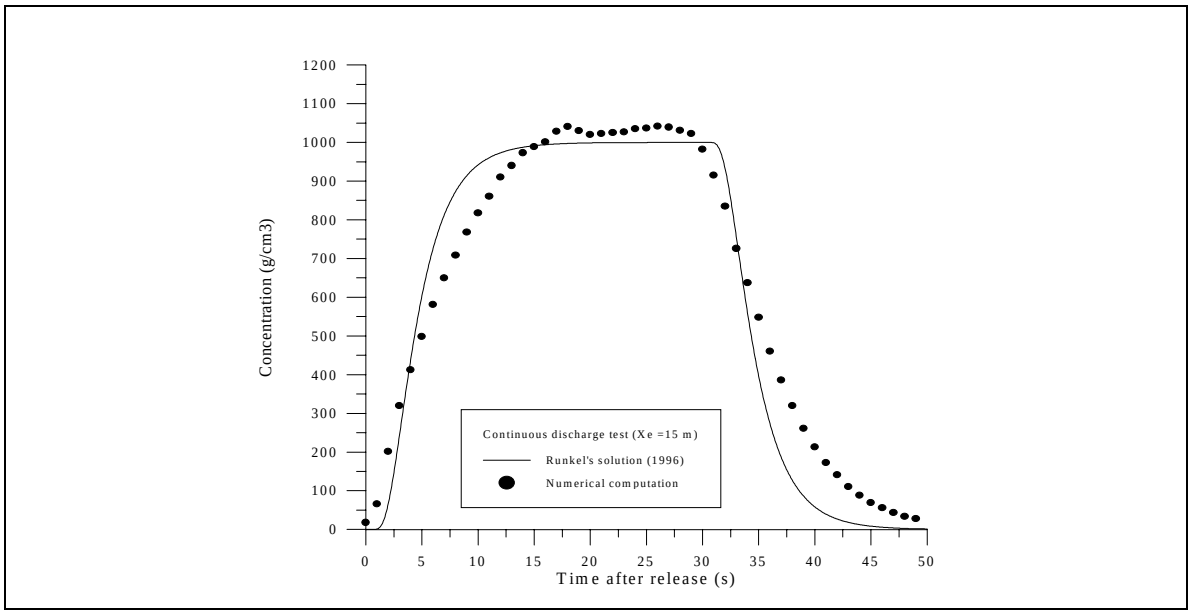

Figure 5.71: Analytical and modelled concentration distribution for the continuous one-dimensional discharge, as a function of time, at a fixed position (x_k = 15), using SPH method, with an integration length of 20 m.

As it can be seen, the larger the integration volume, the smoother the concentration profile at the centre of the distribution is, but also the larger the deviation from the theoretical distribution is near the edges. Like before, the relative error computed using both methods is shown in figure (5.72).

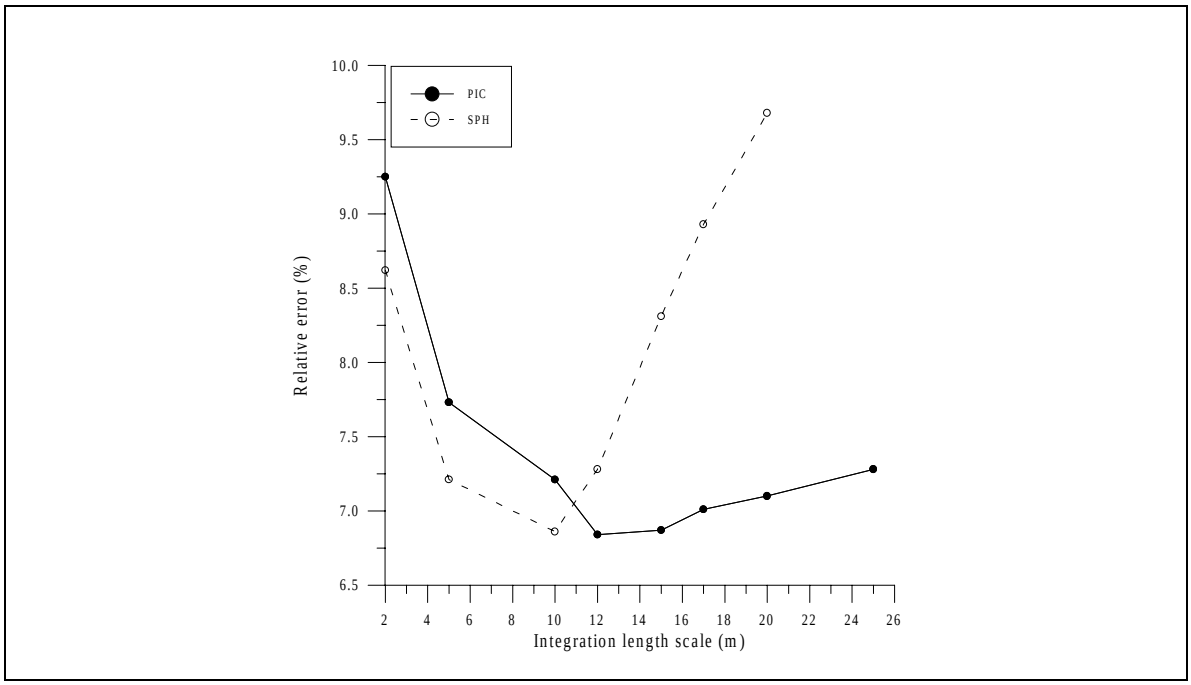

Figure 5.72: Relative error as a function of integration lengthscale, for the BC and the SPH methods.

From the results presented here, it can be concluded that the developed transport model succeeds in reproducing the transport of substances in the conditions specified by these test cases. The comparison of computed and analytical data reveals that the algorithm used to model turbulent diffusion is sufficiently accurate, provided correct values for the diffusivities are supplied; the one-dimensional advective and diffusive transport modelled by LIMMIX also agrees satisfactorily with the analytical solution, as given by Runkel (1996). In all the modelled cases, however, the influence of the integration volume or lengthscale is evident, showing jagged or oversmoothed distributions when the chosen value is too small or too large.

5.4 CONCLUSIONS

The numerical modelling of the general transport equation may be accomplished by using a number of different algorithms, which are generally classified into families according to their description and treatment of the problem. Eulerian and Lagrangian methods solve the relevant equations without making any assumption on the form of the final solution, whereas integral (also called Gaussian) methods solve the equations only along a given characteristic of the problem, and then require a pre-defined variable distribution to obtain values away from the characteristic. Each one of these categories presents a number of strongpoints and disadvantages that make them suitable for a given type of problem, and must be taken into account when a numerical transport model is to be developed.

For the development of the model presented in this thesis, a Lagrangian random walk particle approach has been selected for a number of reasons. The model is to be applied to transport in coastal waters, which is mainly related to marine outfall discharges or sediment load motion; in both cases, it may be important to simultaneously reproduce the transport of two or more different constituents (such as sediment grains of different characteristics), and this is readily accomplished using particle models. Furthermore, although buoyancy forces are not easy to implement in this scheme, a fine vertical resolution may be obtained without the computational burden imposed by a fine 3D mesh, and computer storage memory and CPU-time are reduced by solving the equations only where transport occurs, not over the whole domain.

The different modules representing the relevant transport mechanisms have been developed based on the Lagrangian approach, and then assembled to combine their effects for the overall transport simulation. The advection module includes the effects of hydrodynamic currents and wave-induced mass transport; the turbulent diffusion module is based on a zero-equation turbulent model, using different expressions for the turbulent diffusivities, and relying on a random walk algorithm to account for the complexity of the turbulent motion. For the particular case of sediment transport a settling velocity, based on a semi-empirical formulation, and a “resuspension velocity” are also computed. In the case of buoyant discharges, a buoyancy term is also modelled using semi-empirical equations; if the discharge contains microbiological organisms, a decay module is made available.

Three different algorithms are implemented to map the particle distribution to continuous concentration distributions. The BC algorithm is a box-counting procedure, obtaining concentrations at a given point from the number of particles enclosed inside a box centred at that point; the kBC method is similar to the previous one, but spreads the mass of the particles inside a box across neighbouring boxes, to smooth out concentration fluctuations. The third method, SPH, is based on assuming a pre-defined particle mass distribution in space, and counting each particle’s contribution to the total mass at a given point.

Finally, a set of validation cases, designed to test different modules within the model, are presented. The results show that the overall performance of the transport model is good, closely matching the behaviour predicted by the analytical solutions. It is also seen that a threshold exists for the number of particles used in the simulation, above which the gain in accuracy obtained by increasing the number of particles does not compensate the computational burden (CPU- and memory-wise) that those extra particles introduce. The value of this threshold, however, depends on both the problem to be solved and the computational parameters used in the model run, as the selected mapping algorithm and the prescribed integration length or volume.

In addition, it can be deduced from the results shown in this chapter that all the presented mapping algorithms yield good approximations to the expected solutions, although the kBC algorithm tends to smooth out the concentration distributions too much, and its use is not recommended unless the other methods give excessively jagged distributions. For the BC and SPH methods, given a number of particles, the latter generally performs better than the former, but in both cases the results depend on the chosen integration lengthscale and, in the SPH case, of the selected kernel function. Conversely, less particles are required by the SPH method to achieve an accuracy similar to that of BC, but the possible savings in storage memory and CPU-time consumption introduced because less particles are moved around may vanish due to the higher complexity of the calculations in the mapping module when SPH is used.