Mathematical and Computer Modelling 50 (2009) 1177–1187

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Adaptive stochastic numerical scheme in parallel random walk models for transport problems in shallow water

W.M. Charles [a,*], E. van den Berg [b,1], H.X. Lin [b,1], A.W. Heemink [b,1]

[a] University of Dar-es-salaam, Faculty of Science, Department of Mathematics, P.O. Box 35062, Dar-es-salaam, Tanzania

[b] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Department of Applied Mathematical Analysis, Mekelweg 4, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 21 May 2008
Received in revised form 23 April 2009
Accepted 26 May 2009

Keywords:
Wiener processes
Stochastic differential equation
Random walk model
Variable step size
Parallel processing
Speed up
Efficiency

ABSTRACT

This paper deals with the simulation of transport of pollutants in shallow water using random walk models and develops several computation techniques to speed up the numerical integration of the stochastic differential equations (SDEs). This is achieved by using both random time stepping and parallel processing.

We start by considering a basic stochastic Euler scheme for integration of the diffusion and drift terms of the SDEs, with a strong order 1 in the strong sense. The errors due to this scheme depend on the location of the pollutant; it is dominated by the diffusion term near boundaries, and by the deterministic drift further away from the boundaries. Using a pair of integration schemes, one of strong order 1.5 near the boundary and one of strong order 2.0 elsewhere, we can estimate the error and approximate an optimal step size for a given error tolerance. The resulting algorithm is developed such that it allows for complete flexibility of the step size, while guaranteeing the correct Brownian behaviour.

Modelling pollutants by non-interacting particles enables the use of parallel processing in the simulation. We take advantage of this by implementing the algorithm using the MPI library. The inherent asynchronic nature of the particle simulation, in addition to the parallel processing, makes it difficult to get a coherent picture of the results at any given points. However, by inserting internal synchronisation points in the temporal discretisation, the code allows pollution snapshots and particle counts to be made at times specified by the user.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Environmental management of water areas needs a clear knowledge of the consequences of, e.g., ship accidents and both solid and liquid waste disposal. Therefore, a good environmental control system must allow for the possibility of accurately predicting the dispersion of pollutants if such scenarios arise. Pollutant dispersion is quite often described by the well-known advection–diffusion equations. These partial differential equations are deterministic in nature [1,2]. This Eulerian approach considers the movement of a group of particles by describing what happens at a fixed region in space. Another approach, which describes the dispersion processes, is called the Lagrangian approach and tracks the movement of individual pollutant particles as they move through space in time. The Lagrangian approach is stochastic in nature and is often called a particle model.

* Corresponding author. Tel.: +255 222410388.
E-mail addresses: maheraii@yahoo.com, cwmahera@maths.udsm.ac.tz (W.M. Charles), vandenberg_ewout@yahoo.co.uk (E. van den Berg), haix_lin@yahoo.com (H.X. Lin), A.W.Heemink@ewi.tudelft.nl (A.W. Heemink).
1 Tel.: +31 15 2787228; fax: +31 15 2787295.

0895-7177/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2009.05.034

The advantage of the Lagrangian approach over the Eulerian ones is that it models well the steep concentration gradient for particles that are independent of one another. Moreover, unlike in the Eulerian approach, particle tracking is independent of the domain, and the concentration can not be negative [3–5]. However, when domain is large, a large number of particle is required to accurately predict the dispersion of the concentration of pollutants. This implies that particle models require more efficient integration schemes.

The work in this article aims for the design and implementing of a more accurate and efficient scheme used in the particle model. It was noted by Stijnen et al. [6] that the simulation of pollutant transport in shallow waters using random walk models requires small step sizes to stably integrate in highly irregular areas. Traditional random walk models are based on fixed step size implementation of numerical methods and are thus forced to choose a small step size throughout. This is unnecessary in regular areas where the integration error is very small and larger time steps can be used. In such situations, the use of adaptive numerical schemes is more appropriate.

In their work Gaines and Lyons [7] and Burrage and Burrage [8] introduced a variable time stepping procedure for the pathwise (strong) numerical integration of a system of stochastic differential equations. In this paper we extend the work from [9], and use pairs of adaptive schemes to determine appropriate step sizes. Each pair consists of a lower and higher order scheme and, by evaluating the performance of each scheme, it is possible to decide whether higher order schemes, or small step sizes are needed.

For the numerical integration of the pollutant transport models we use an explicit order 1 strong scheme. The displacement of a pollutant particle is due to drift and diffusion terms. Depending on the location of a particle, the error from the first order scheme arises predominantly from either the drift or diffusion term. Near boundaries it is the diffusion term that dominates and we pair up the first order scheme with an explicit order 1.5 strong scheme. Further away from the boundary, the drift term becomes dominant and the 1.5 order scheme becomes order 1 strong. Consequently, it is necessary to concentrate on the drift term leading us to derive an explicit order 2 strong scheme that concentrates on this drift error while approximating the diffusion term using a linear stochastic differential equation with additive noise.

Outside particle models, the concept of adaptive schemes by mesh refining in Eulerian methods has been used in [10]. Unlike these methods, random walk models do not suffer from numerical diffusion in the source points [5,4], however, they do potentially require large numbers of particles [11]. Fortunately, since particles are simulated independent from one another, parallel processing techniques can be applied naturally. This approach was used to speed up the implementation of the model discussed in this paper.

This remainder of this paper is organised as follows. Section 2 discusses the set of stochastic differential equations governing the transport of pollutants in shallow water. The concept of the higher order strong adaptive schemes for transport problems in shallow water as well their implementation is discussed in Section 3. The procedure for determining the variable step sizes is described in Section 3.4. The parallel implementation of the adaptive scheme is given in Section 4 and experimental results and concluding remarks appear in Sections 5.1 and 6.

2. The random walk model for pollution transport

Random walk models describe the behaviour of a large number of particles, each representing a part of the total mass of pollutant [12,1,13,14]. In such particle models, pollution is described by a set of particles and the position of individual particles is described by means of stochastic differential equations (SDEs). In particular, the three-dimensional particle position at time t by $P_t = [X_t, Y_t, Z_t]$, is given by the (Îto) stochastic differential equation [15–17]

$$\mathrm{d}P = f(t, P)\mathrm{d}t + \sigma(t, P)\mathrm{d}W(t), \quad t \geq 0,$$

where $f(t, P)\mathrm{d}t$ is a vector-valued drift term, $\sigma(t, P)$ is a matrix function, and $\mathrm{d}W(t)$ represents the Brownian motion process. The movement of individual particles is subject to a deterministic part, mainly given by the flow field; and a stochastic part, which represents the turbulence in the fluid, and is modelled by the diffusion part of the stochastic differential equations [5]. Using the particle positions, a probability density function, which is related to the ensemble mean concentration, can be approximated [1,14,18].

The particle model derived in this paper is a vertically integrated model, which integrates over depth and results in a two-dimensional model:

$$\begin{aligned}
\mathrm{d}X_t &\overset{\text{Itô}}{=} \left[U(x,y) + \frac{D_{XX}(x,y)}{H(x,y)}\left(\frac{\partial H(x,y)}{\partial x}\right) + \frac{\partial D_{XX}(x,y)}{\partial x}\right]\mathrm{d}t + \sqrt{2D_{XX}(x,y)}\mathrm{d}W^x \\
\mathrm{d}Y_t &\overset{\text{Itô}}{=} \left[V(x,y) + \frac{D_{YY}(x,y)}{H(x,y)}\left(\frac{\partial H(x,y)}{\partial y}\right) + \frac{\partial D_{YY}(x,y)}{\partial y}\right]\mathrm{d}t + \sqrt{2D_{YY}(x,y)}\mathrm{d}W^y.
\end{aligned} \tag{1}$$

Here, (X_t, Y_t) is the position of a particle, which is assumed to be a Markov process, $W^x(t)$ and $W^y(t)$ are Gaussian–Wiener processes [11], $(U(x, y), V(x, y))^{\mathrm{T}}$ are flow velocities, and $H(x, y)$ is the total water depth. The latter two quantities, flow velocity and water depth, are assumed to be inputs to the model. In our case they are computed over a fixed grid using a hydrodynamic model known as WAQUA [19], and extended to the entire domain by using (linear) interpolation techniques.

The two-dimensional random walk model Eq. (1) were developed by Heemink [5] and shown to be consistent with the two-dimensional advection–diffusion equation by deriving the Fokker–Planck equations. In this article we modify this model by using a different dispersion coefficient function to model pollutants dispersion in shallow waters. In addition, we design the horizontal dispersion coefficient functions $D_{XX}(x, y)$, and $D_{YY}(x, y)$ in the x and y direction respectively. We assume that $D_{XY}(x, y) = D_{YX}(x, y) = 0$ in the dispersion tensor and that

$$D_{XX}(x, y) = \frac{D11}{1 + e^{-(((x-xb)^2+(y-yb)^2)-K^2)}} \left\{1 + \left([1 + e^{K^2}]\cos(\alpha) - 1\right) e^{-((x-x_b)^2+(y-y_b)^2)}\right\} \tag{2}$$

$$D_{YY}(x, y) = \frac{D22}{1 + e^{-(((x-xb)^2+(y-yb)^2)-K^2)}} \left\{1 + \left([1 + e^{K^2}]\sin(\alpha) - 1\right) e^{-((x-x_b)^2+(y-y_b)^2)}\right\}, \tag{3}$$

where $D11$ and $D22$ are the horizontal dispersion parameters, (xb, yb) is intersection point on the boundary between the line from (x, y) perpendicular to the boundary, $K \geq 0$ is a parameter modelling the decrease of diffusion coefficient near the boundary and α is the assumed to be the angle between the boundary, and x or y direction. Furthermore we have

$$\underline{e}_1^T \underline{c} = \|\underline{e}_1\| \|\underline{c}\| \cos(\alpha), \qquad \sin(\alpha) = \frac{|c_2|}{\sqrt{c_1^2 + c_2^2}},$$

with $\underline{e}_k$ the kth column of the identity matrix. Finally, $\underline{c} = (c_1, c_2)^{\mathrm{T}}$ is a direction vector along the side of a given boundary cell.

In this article the diffusion functions are designed in such a way that $D_{XX}(x, y)$ and $D_{YY}(x, y)$ model the dispersion as realistically as possible. This means, for instance, that the dispersion decreases near the boundary and that no particle will cross a physical boundary. By setting $D_{XY} = D_{YX} = 0$ we assume for simplicity that the dispersion tensor is an ellipsoid with main axes in the x and y direction only. This assumption is not essential to the derivation and it would not be hard to incorporate nonzero coefficients [20].

3. Adaptive scheme with higher order strong for pollution transport

In this paper we make use of strong convergence of pairs of numerical schemes to approximate the error of the scheme at each step. To achieve this, we have developed three numerical schemes for transport of pollutant in shallow waters: one explicit order 1 strong scheme, discussed in Section 3.1, and two higher order schemes, formulated in Sections 3.2 and 3.3. These higher order strong numerical schemes avoid the use of derivatives of the drift and diffusion terms and are of order 1.5 strong and 2.0 strong, respectively [21]. We only give a brief overview of the schemes; an interested reader is referred to [21,11]. Based on the error information at each step the optimal time step size for adaptive schemes is determined, as explained in Section 3.4.

3.1. An explicit order 1 strong scheme

The explicit order 1 scheme for transport of pollutants in shallow waters is as follows:

$$\begin{aligned} X_{n+1} \overset{\text{Itô}}{=} X_n &+ \left[U(x, y) + \frac{D_{XX}(X_n, Y_n)}{H(x, y)} \frac{\partial H(x, y)}{\partial x} + \frac{\partial D_{XX}(X_n, Y_n)}{\partial x}\right] \Delta t_n \\ &+ \frac{(\Delta W_n^x)^2 - \Delta t_n}{2\sqrt{\Delta t_n}} \left[\sqrt{2D_{XX}(X_{n+1}^{*+1}, Y_{n+1}^{*+1})} - \sqrt{2D_{XX}(X_n, Y_n)}\right] + \sqrt{2D_{XX}(X_n, Y_n)} \Delta W_n^x, \end{aligned} \tag{4}$$

where $W_n^x = W^x(t_n)$, $\Delta W_n^x = W_{n+1}^x - W_n^x$ is an independent increment of a Wiener process for the time interval $[t_n, t_{n+1}]$, $n = 0, 1, \ldots$. The expression for Y_{n+1} is similar to the above equation, with X_n in the first right hand side term replaced by Y_n, all $D_{XX}(\cdot, \cdot)$ terms substituted by $D_{YY}(\cdot, \cdot)$ and the superscripts x modified to y. We further have support vector

$$X_{n+1}^{*+1} = X_n + a^1(X_n, Y_n)\Delta t_n + \sqrt{2D_{XX}(X_n, Y_n)\Delta t_n}.$$

and similarly for Y_{n+1}^{*+1} along y direction. The drift along the x direction is given by

$$a^1(X_n, Y_n) = \left[U(x, y) + \frac{D_{XX}(X_n, Y_n)}{H(x, y)} \left(\frac{\partial H(x, y)}{\partial x}\right) + \frac{\partial D_{XX}(X_n, Y_n)}{\partial x}\right], \tag{5}$$

and along the y direction we have, likewise

$$a^2(X_n, Y_n) = \left[V(x, y) + \frac{D_{YY}(X_n, Y_n)}{H(x, y)} \left(\frac{\partial H(x, y)}{\partial y}\right) + \frac{\partial D_{YY}(X_n, Y_n)}{\partial y}\right]. \tag{6}$$

3.2. An explicit order 1.5 strong scheme

Besides the explicit order 1 strong scheme we also derived an order 1.5 scheme (see [21]) which avoids evaluating derivatives of the drift and diffusion terms:

$$\begin{aligned}
X_{n+1} \overset{\text{Itô}}{=} X_n &+ \left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) - a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} \times \frac{\Delta t_n}{4}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right) \\
&+ \frac{\Delta t_n}{4}\left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) + a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} + \sqrt{2D_{XX}(X_n, Y_n)}\Delta W_n^x \\
&+ \frac{1}{4\sqrt{\Delta t}}\left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\}\left[(\Delta W_n^x)^2 - \Delta t_n\right] \\
&+ \left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - 2\sqrt{2D_{XX}(X_n, Y_n)} + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\} \\
&\times \left\{\Delta W_n^x - \frac{1}{2}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right)\sqrt{\Delta t_n}\right\} \\
&+ \left[\sqrt{2D_{XX}(X_{n+1}^{*+\phi}, Y_{n+1}^{*+\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*-\phi}, Y_{n+1}^{*-\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}\right. \\
&\left. + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right] \times \frac{1}{4\Delta t}\left\{\frac{1}{3}(\Delta W_n^x)^2 - \Delta t_n\right\}\Delta W_n^x. \qquad (7)
\end{aligned}$$

The expression for Y_{n+1} is again similar, with the first right hand side term, X_n, replaced by Y_n, all terms $D_{XX}(\cdot,\cdot)$ changed to $D_{YY}(\cdot,\cdot)$, and the superscripts x modified to y for W and R, likewise substitute the $1+$ and $1-$ superscripts for a by $2+$ and $2-$. The terms $R_{n,1}$ and $R_{n,2}$ denote i.i.d. normally distributed random variables. Using the shorthand notation of $\oplus$ for either $+$ or $-$, the following supporting vectors (used in Eq. (7)) are defined by

$$\begin{aligned}
X_{n+1}^{*\oplus z} &= X_n + \frac{1}{2}a_n^1(X_n, Y_n)\Delta t_n \oplus \sqrt{2D_{XX}(X_n, Y_n)\Delta t_n} \\
X_{n+1}^{*\oplus\phi} &= X_{n+1}^{*\oplus z} \oplus \sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})\Delta t_n}.
\end{aligned}$$

For the expressions $Y_{n+1}^{*+z}, Y_{n+1}^{*-z}, Y_{n+1}^{*+\phi}$, and $Y_{n+1}^{*-\phi}$ change the first term X to Y, and replace a_n^1 and D_{XX} by a_n^2 and D_{YY} respectively. $R_{n,2}^x$ generates random numbers that are uniformly distributed. By applying Eq. (5), we get

$$\begin{aligned}
a^{1+}\left(X_{n+1}^{*+z}, Y_{n+1}^{*+z}\right) &= \left[U(x,y) + \frac{D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}{H(x,y)}\frac{\partial H(x,y)}{\partial x} + \frac{\partial D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}{\partial x}\right] \\
a^{1-}\left(X_{n+1}^{*-z}, Y_{n+1}^{*-z}\right) &= \left[U(x,y) + \frac{D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}{H(x,y)}\frac{\partial H(x,y)}{\partial x} + \frac{\partial D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}{\partial x}\right],
\end{aligned}$$

and likewise for $a^{2+}\left(X_{n+1}^{*+z}, Y_{n+1}^{*+z}\right)$ and $a^{2-}\left(X_{n+1}^{*-z}, Y_{n+1}^{*-z}\right)$.

The functions $D_{XX}(\cdot,\cdot)$ and $D_{YY}(\cdot,\cdot)$ are constant away from the boundary but approach zero towards the boundary. In this region, the error of the numerical scheme is mainly due the deterministic part of the model. To deal with this, we need a scheme with a higher strong order of convergence. We now describe such a strong order 2 scheme.

3.3. An explicit order 2 strong scheme

The derivative-free order 1 and 1.5 strong schemes discussed above were obtained by replacing the derivatives with finite differences [21]. Although this technique works well for low order explicit schemes, it results in increasingly complicated formulae as the order increases. To somewhat reduce the complexity we take advantage of the structure of the stochastic differential equations and assume that the noise is additive, i.e., that the diffusion coefficient is not space dependent. This way, we obtain a relatively simple explicit higher order scheme for the transport of pollutants is shallow waters.

Under additive noise, the Itô and Stratonovich concepts coincide. For the derivation of the explicit order 2 strong scheme for the additive noise we have chosen to use the Stratonovich interpretation. Taking into account the independence of time of D_{xx} and D_{yy}, the resulting scheme is as follows:

$$X_{n+1} \overset{\text{Strat}}{=} X_n + \frac{1}{2}\left\{\underline{a}^1\left(X_{n+1}^+, Y_{n+1}^+\right) + \underline{a}^1\left(X_{n+1}^-, Y_{n+1}^-\right)\right\}\Delta t_n + \sqrt{2D_{XX}(X_n, Y_n)}\Delta W_n^x \qquad (8)$$

$$Y_{n+1} \overset{\text{Strat}}{=} Y_n + \frac{1}{2}\left\{\underline{a}^2\left(X_{n+1}^+, Y_{n+1}^+\right) + \underline{a}^2\left(X_{n+1}^-, Y_{n+1}^-\right)\right\}\Delta t_n + \sqrt{2D_{YY}(X_n, Y_n)}\Delta W_n^y. \qquad (9)$$

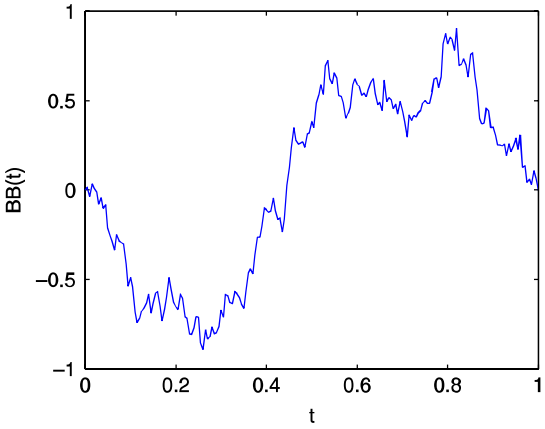


Fig. 1. A sample Brownian bridge path.

The supporting vectors are defined by

$$X_{n+1}^{\oplus} = X_n + \frac{1}{2}\underline{a}^1(X_n, Y_n)\Delta t_n + \frac{1}{\Delta t_n}\sqrt{2D_{XX}(X_n, Y_n)}\left\{\Delta M_n^x \oplus \sqrt{2J_{(1,1,0)}^{x,p}\Delta t_n - (\Delta M_n^x)^2}\right\} \tag{10}$$

$$Y_{n+1}^{\oplus} = Y_n + \frac{1}{2}\underline{a}^2(X_n, Y_n)\Delta t_n + \frac{1}{\Delta t_n}\sqrt{2D_{YY}(X_n, Y_n)}\left\{\Delta M_n^y \oplus \sqrt{2J_{(1,1,0)}^{y,p}\Delta t_n - (\Delta M_n^y)^2}\right\}, \tag{11}$$

where $\oplus$ denotes the plus or minus operator. The definition of $\underline{a}^1(X, Y)$ is obtained by using the Itô–Stratonovich transformation [11] of Eq. (5), yielding

$$\underline{a}^1(X, Y) = \left[U(x, y) + \frac{D_{XX}(X, Y)}{H(x, y)}\left(\frac{\partial H(x, y)}{\partial x}\right) + \frac{1}{2}\frac{\partial D_{XX}(X, Y)}{\partial x}\right].$$

$$\underline{a}^2(X, Y) = \left[V(x, y) + \frac{D_{YY}(X, Y)}{H(x, y)}\left(\frac{\partial H(x, y)}{\partial y}\right) + \frac{1}{2}\frac{\partial D_{YY}(X, Y)}{\partial y}\right].$$

Higher order schemes, such as those of order 2, require the approximation of multiple higher Stratonovich stochastic integrals ($J_{(1,1,0)}^p$) (see Eq. (12)). Unfortunately, these cannot always be expressed in terms of simpler stochastic integrals, especially when the Wiener process is multi-dimensional. However, with the aid of a method for multiple Stratonovich based on Karhunen–Loève or random Fourier series expansion of the Wiener process (for details, see [11]), we can approximate the multiple integrals. This introduces the Brownian bridge (see [11]) into our model. The Brownian bridge is a restricted Wiener process (hence also referred to as the "tied down" Wiener process) that passes through known points at $t = 0$ and $t = T$ and is given by $\left\{W_t - \frac{t}{T}W_T, 0 \le t \le T\right\}$. This can be done by generating an unconstrained (standard) Wiener process which is then linearly scaled in order to meet the required end points. An example of a Brownian bridge path, which is tied down to zero at beginning and at the end points, is shown in Fig. 1.

Following Karhunen–Loève [11], we define the random variables a_r^x and b_r^x by

$$a_r^x = \frac{2}{\Delta t}\int_0^{\Delta t}\left(W_s^x - \frac{s}{\Delta t}W_{\Delta t}^x\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)\mathrm{d}s$$

and

$$b_r^x = \frac{2}{\Delta t}\int_0^{\Delta t}\left(W_s^x - \frac{s}{\Delta t}W_{\Delta t}^x\right)\sin\left(\frac{2r\pi s}{\Delta t}\right)\mathrm{d}s, \quad r = 1, 2, \ldots.$$

Likewise, a_r^y and b_r^y are obtained by replacing the x superscripts by y. (In the remainder of this section we will silently assume this convention, unless otherwise specified.) It is known that, for $r \ge 1$ these variables have an $\mathcal{N}\left[0, \frac{\Delta t}{2\pi^2 r^2}\right]$ distribution [21] and are differentiable sample paths on the interval $[0, T]$. Now, let $\zeta_r^x, \xi^x, \zeta_r^y, \xi^y, \eta_r^x, \eta_r^y, \phi_p^x$, and ϕ_p^y denote independent random variables [11] defined, for $r = 1, 2, \ldots$ and $p = 1, 2, \ldots$, by

$$\xi^x = \frac{1}{\sqrt{\Delta t}}W_{\Delta t}^x, \qquad \zeta_r^x = \sqrt{\frac{2}{\Delta t}}\pi r a_r^x, \qquad \eta_r^x = \sqrt{\frac{2}{\Delta t}}\pi r b_r^x$$

$$\mu_p^x = \frac{1}{\sqrt{\Delta t\rho_p}}\sum_{r=p+1}^{\infty} a_r^x, \qquad \phi_p^x = \frac{1}{\sqrt{\Delta t\beta_p}}\sum_{r=p+1}^{\infty}\frac{1}{r}b_r^x$$

$$\mu_p^y = \frac{1}{\sqrt{\Delta t\rho_p}}\sum_{r=p+1}^{\infty} a_r^y, \qquad \phi_p^y = \frac{1}{\sqrt{\Delta t\beta_p}}\sum_{r=p+1}^{\infty}\frac{1}{r}b_r^y.$$

The variance of $\hat{\mu}_p^x = \sqrt{\Delta t \rho_p}\mu_p^x$ can be computed by noting that the variance of a_r^x is given by $\text{var}[a_r^x] = \Delta t/(2\pi^2 r^2)$ [21]. Since the variance of two independent Gaussian variables equals the sum of variances and using the fact that $\sum_{r=1}^{\infty} 1/r^2 = \pi^2/6$ and $\sum_{r=1}^{\infty} 1/r^4 = \pi^4/90$, we find that

$$a_0^x = -\frac{1}{\pi}\sqrt{2\Delta t}\sum_{r=1}^{p}\frac{1}{r}\zeta_r^x - 2\sqrt{\Delta t \cdot \rho_p}\mu_p^x, \quad \rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2}.$$

Using the definition of a_r^x, a_r^y and denoting p the truncation index in the approximation of multiple integrals, we then define

$$B^x = \sqrt{\frac{\Delta t}{2}}\sum_{r=1}^{p}\frac{1}{r^2}\eta_r^x + \sqrt{\Delta t \beta_p}\phi_p^x, \quad \beta_p = \frac{\pi^2}{180} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^4}.$$

Furthermore, we have

$$\Delta M_n^x = \frac{1}{2}\Delta t\left[\sqrt{\Delta t}\xi^x + a_0^x\right], \qquad C_{x,x}^p = -\frac{1}{2\pi^2}\sum_{r,l=1_{r\neq l}}^{p}\frac{r}{r^2-l^2}\left\{\frac{1}{l}\zeta_r^x\zeta_l^x - \frac{l}{r}\eta_r^x\eta_l^x\right\}$$

and similar for ΔM_n^y and $C_{y,y}^p$ and with superscripts changed from x to y. Using these random variables it follows from a lengthy computation, that we can approximate the multiple integral by:

$$J_{(1,1,0)}^{x,p} = \frac{1}{6}(\Delta t)^2(\xi^x)^2 + \frac{1}{4}\Delta t(a_0^x)^2 - \frac{1}{2\pi}(\Delta t)^{\frac{3}{2}}\xi^x B^x + \frac{1}{4}(\Delta t)^{\frac{3}{2}}a_0^x\xi^x - (\Delta t)^2 C_{x,x}^p, \tag{12}$$

$$J_{(1,1,0)}^{y,p} = \frac{1}{6}(\Delta t)^2(\xi^y)^2 + \frac{1}{4}\Delta t(a_0^y)^2 - \frac{1}{2\pi}(\Delta t)^{\frac{3}{2}}\xi^y B^y + \frac{1}{4}(\Delta t)^{\frac{3}{2}}a_0^y\xi^y - (\Delta t)^2 C_{y,y}^p, \tag{13}$$

$J_{(1,1,0)}^{x,p}$ is an approximation of $J_{(1,1,0)}^x$ and it is known [11] that $J_{(1,1,0)}^x \geq \frac{(\Delta M^x)^2}{2\Delta t_n}$ always. If it turns out $J_{(1,1,0)}^{x,p} < \frac{(\Delta M^x)^2}{2\Delta t_n}$, we take ΔM^x as the better approximation for $J_{(1,1,0)}$. Similarly for $J_{(1,1,0)}^{y,p}$ and finally the evaluations of Eqs. (8) and (9) can take place.

3.4. Determination of variable time step sizes

Given a current particle position (X_n, Y_n) we can apply the explicit order 1 string scheme in Eq. (4) to obtain an approximate solution, $(\hat{X}_{n+1}, \hat{Y}_{n+1})$, for step $n+1$. Similarly, depending on whether the current location is near a boundary or not, we can use the explicit order 1.5 (Eq. (7)) or 2.0 (Eqs. (8) and (9)) schemes to obtain the reference location, (X_{n+1}^0, Y_{n+1}^0). Based on these two positions, we estimate the absolute error, as explained in [8], and from that the final step length. In either case the higher order reference scheme is used to advance the numerical computation to the next time step. Note that all of this is done individually for each particle.

Let τ_i be the tolerance accepted for the ith component. Then an error estimate of the order $q + \frac{1}{2}$ for the two-dimensional adaptive random walk model is given by:

$$\epsilon = \sqrt{\frac{1}{2}\left(\left|\frac{X_{n+1,1}^0 - \hat{X}_{1n+1,1}}{\tau_1}\right| + \left|\frac{Y_{n+1,2}^0 - \hat{Y}_{1n+1,2}}{\tau_2}\right|\right)}, \tag{14}$$

where q is considered to be either order $\hat{o}$ or order o. It is desirable that $|X_{n+1,1}^0 - \hat{X}_{n+1,1}| \approx \tau_1$ and $|Y_{n+1,2}^0 - \hat{Y}_{n+1,2}| \approx \tau_2$. Therefore, a step is rejected whenever $\epsilon > 1$ and a new step size $\Delta t := \Delta t/\sqrt{\epsilon}$ is tried, until the desired accuracy is attained.

For efficient implementation of the adaptive scheme using, an optimal step size can be decreased by a safety factor, e.g., $\gamma = 0.8$. This avoids unnecessary computations by ensuring the step size does not increase or decrease too quickly [8]:

$$\Delta t := \Delta t \cdot \min\left(\gamma_{\max}, \max(\gamma_{\min}, \gamma/\sqrt{\epsilon})\right), \tag{15}$$

with $\gamma_{\max}$ and $\gamma_{\min}$ respectively the maximal and minimal step size scaling factors allowed [8]. In case the initial time step is unnecessarily small, Eq. (15) is used to enlarge it. Finally, our approach avoids uncontrolled jumps in the step size by setting the final step length to

$$\Delta t_n = \max\left(\Delta t, 0.9 \cdot \Delta t_{n-1}\right).$$

4. Implementation of variable time stepping in random walk models

The implementation of an adaptive scheme differs substantially from one with a fixed step size in that it is no longer possible to have a single major loop governing the time and taking single steps of fixed size [22]. Instead, the current time differs between the particles and, in addition to the coordinates, each particle now needs to have a local time associated with it. This concept of local time introduces a high level of asynchronicity into the model, making it harder to gather information about the model at a given point in time. Before discussing how this problem is dealt with, we take a brief look at the global architecture of the program.

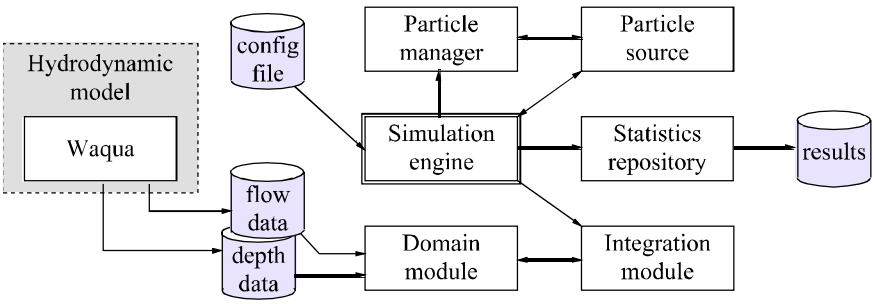


Fig. 2. An overview of the implementation of the parallel adaptive random walk model for pollution dispersion in shallow water.

As shown in Fig. 2, the program consists of a number of modules, the most important of which are:

(1) **Simulation engine**; the central component of the application orchestrating all other parts
(2) **Domain module**; deals with the loading and interpolation of flow and depth information of the domain
(3) **Statistics repository**; collects all desired information about the model during the run
(4) **Integration module**; performs the numerical integration of the stochastic differential equations on a set of particles.

The first three of these modules define events that arise at certain times during the simulation. The simulation engine has a stop simulation event that arises when the final simulation time has been reached. The domain module needs to update the current flow and depth information at regular intervals. These data were computed at regular time intervals of 60 min each, using a hydrodynamic model called WAQUA [19]. Finally, the statistics repository may request a snapshot of the current situation to be taken at user defined times. As shown in the pseudocode of the simulation engine in Algorithm 1, the events define the time intervals between which the integration module simulates the particles. Based on the particle's current time (which can be larger than the last event time, e.g., if a particle was determined to flow in from across the boundary at that time) it can use whatever step sizes are deemed appropriate until either the event time has been reached, or the particle has ceased to exist, for instance when they flow out of the open boundaries or degrade naturally. With this mechanism the synchronicity, albeit at a coarser and irregular level, is restored; whenever an event happens all particle information is guaranteed to be valid at that point in time.

Algorithm 1. Simplified representation of the main event loop of the simulation engine.

```
1:  Set up and configure the simulation
2:  t ← t_0
3:  loop
4:     Get event e from queue
5:     while t(e) = t do
6:        Handle event e and add any new events to the queue
7:        if type(e) = quit then
8:           Raise quit flag
9:        end if
10:       Get event e from queue
11:    end while
12:    Exit main loop if quit flag is set
13:    Generate all new particles until t(e)
14:    Integrate all particles from t until t(e)
15:    Handle event e and add any new events to the queue
16:    t ← t(e)
17: end loop
18: Finalise simulation
```

5. Experiments of parallel processing in the adaptive scheme

The particles in the random walk model do not interact with one another and can therefore be simulated completely independent. As such, the simulation lends itself extremely well for parallel processing with each processor simulating an equal fraction of all particles. This means that instead of doing domain decomposition, we use workload distribution.

In this section, we discuss parallel experiments for the prediction of pollutants dispersion, carried out on a distributed memory parallel architecture called DAS-2 [23]. This system is a wide-area distributed system with 200 dual processor nodes. More detailed information can be found in [23].

We ran two sets of experiments; one on a simple artificial domain and a second one on a realistic model of the Dutch coastal waters.

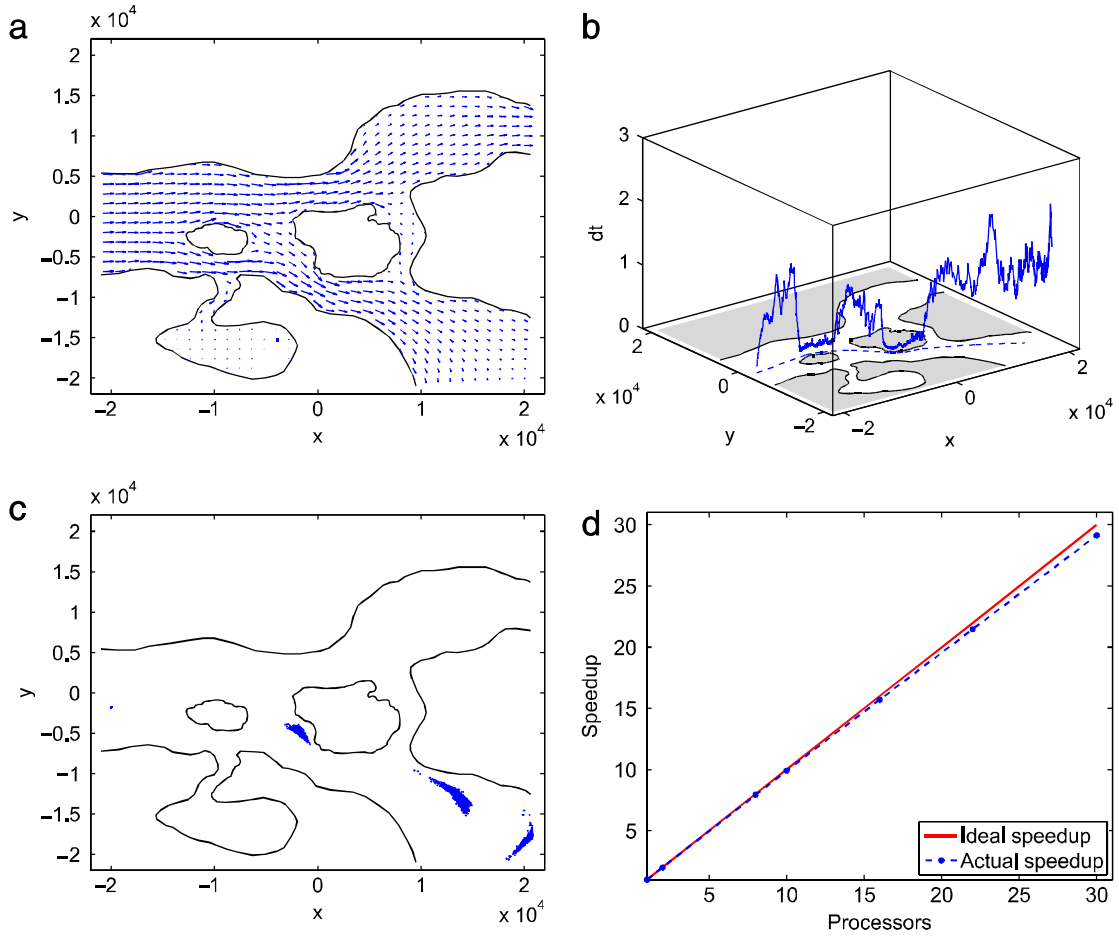


Fig. 3. Plots of (a) the artificial domain, (b) variation of the step size along one track, (c) snapshots of the particle positions at times $t = 0$, 300, 600, and 900 s, and (d) the ideal and actual speed up curves, based on simulation with 100,000 particles.

5.1. Simulation results on the artificial domain

For the first set of experiments we look at a simple artificial domain consisting of a river flowing from the west to the east, a lake and two islands, as shown in Fig. 3(a). In the experiments 2000 particles were released at coordinate $(-20, -1.8)$ at time $t = 0$. At every multiple of 300 time units, a snapshot was taken along with a number of particle traces with location and step size stored at each iteration step. Fig. 3(b) shows one such trace plotted in the horizontal plane with the associated step sizes plotted on the vertical axis. It is clear to see, and in line with our expectations, that the step size is small when the particle approaches and bends along the boundaries whereas it is allowed to grow larger in open water with smoothly varying flow. The maximum step size at $t = 0$ is chosen relatively small which explains the rapid initial growth of the step size. Fig. 3(c) shows four snapshots, taken at times $t = 0$, 300, 600, and 900 respectively, illustrating how the particle cloud slowly diffuses as it flows down the stream. At time $t = 900$ most particles have flowed out of the domain at the south-east.

For the parallel processing experiment we increased the number of particles from 2000 to 100,000 and used up to 30 processors to simulate the model. The measured speed up, along with the ideal curve is plotted in Fig. 3(d). When using 30 processors the speed up is approximately 29.38, which is quite near to the ideal value of 30 (the endpoint of the solid line). The parameters used in the simulation were $\gamma = 0.8$, $\gamma_{\min} = 0.6$, $\gamma_{\max} = 1.1$. Detection of a nearby boundary was done by considering the box of 6 by 6 grid cells around the particle, i.e. with a radius of 3. A maximum of 30 consecutive Brownian bridge refinements per iteration were allowed in this experiment.

Fig. 4(a) plots the step size versus the iteration number. In particular it shows that while fixing the step size to around 0.1 would suffice in most regions, this size would be overly pessimistic in other regions where step sizes up to 2 can be taken. Since the relation between step size and processing time is approximately linear, this implies that in those regions we can easily obtain a speed up of 10–15. For the entire simulation, this amounts to a reduction of the number of integration steps by about a factor five, per particle. In part (b) of the same plot we counted the number of steps taken for different intervals of dt and computed which percentage of the total simulation time was covered by those steps (i.e., by summing the dt values and dividing by the total simulated time). This shows that only about 4% of the time is covered by the smallest time step and a total of 20% in the smallest three bars, leaving around 80% of the total simulation time covered by larger steps. The construction of the adaptive method ensures that the same level of accuracy is attained while allowing these larger step sizes.

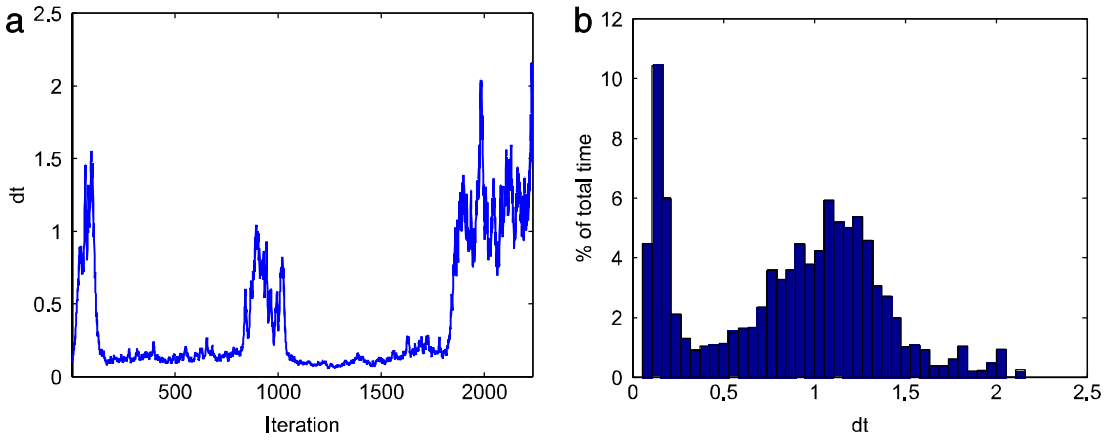


Fig. 4. (a) Value of dt as a function of iteration for a single track (b) the percentage of total time covered by different ranges of dt values.

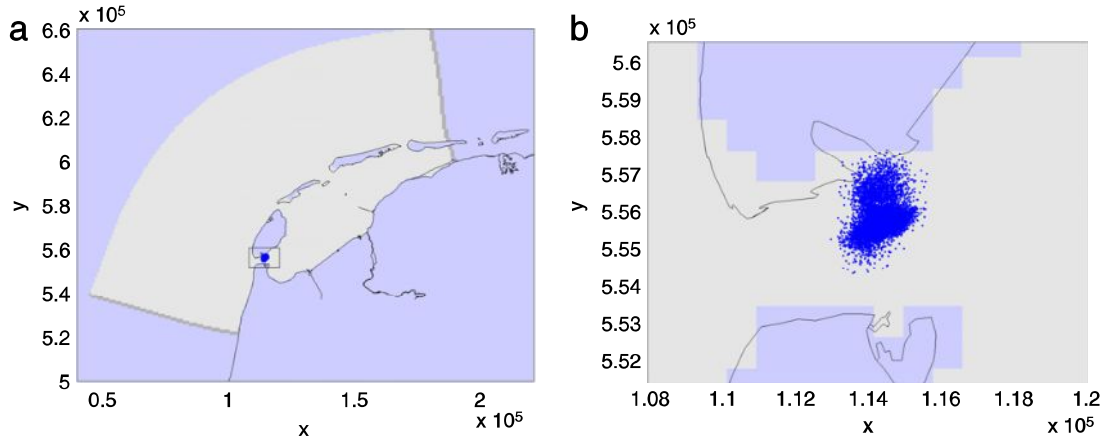


Fig. 5. Snapshot of particles taken $t = 40{,}000$, close-up on the right with shaded blocks indicating land.

5.2. Simulation results on the Dutch coastal waters

In the second set of experiments we study the adaptive random walk model applied to pollutant dispersion using real velocity flow fields and water depth data of a part of the Dutch North Sea. The grid consists of 201 by 225 cells of size 800 by 800 m. At $t = 0$, a number of particles is released at location $(1.14 \times 10^5, 5.56 \times 10^5)$, representing the source of pollution. Their position after 40,000 s is as shown in Fig. 5. The profile of the step size as a fraction of total amount of time for a single particle is plotted in Fig. 6(a). Due to the scale of the problem the effect of the boundaries on the step size is not pronounced as clearly as in the artificial domain.

For parallel computing we first used 400,000 particles and measured of speed up and efficiency for up to 25 processors. For large enough numbers of particles and long enough simulation times, the stochastic nature of the model furthermore automatically guarantees that a good overall load balance over the processors is maintained, even though at given times some processors may be busier than others.

In another experiment using 20,000 particles, a speed up of 24.55 was attained (see Fig. 6(d)). Both cases show that the speed up scales very well, although the speed up in the second experiment is somewhat smaller than in the first experiment. This is caused by the relative increase of overhead in loading the data files, and, to a lesser extent, because of the disparity between processor load which becomes more pronounced as the total amount of work decreases.

6. Concluding remarks

In this paper we develop a particle model with adaptive time stepping for the simulation of pollutant transport in shallow waters and describe an efficient implementation. By tracking the location and step size of individual particles during the simulation we show that the adaptive model effectively chooses step sizes according to position within the domain.

The implementation introduces synchronisation points in the simulation time line to overcome the inherent asynchronicity in temporal position of the particles. This allows us to take snapshots of the particles at arbitrary points in the simulation and visualise the spreading of pollutant over time. The implementation further takes advantage of the fact that each particle can be simulated independently and uses parallel processing to speed up the simulation time. Experiments show that the parallel implementation scales very well with the number of processors used.

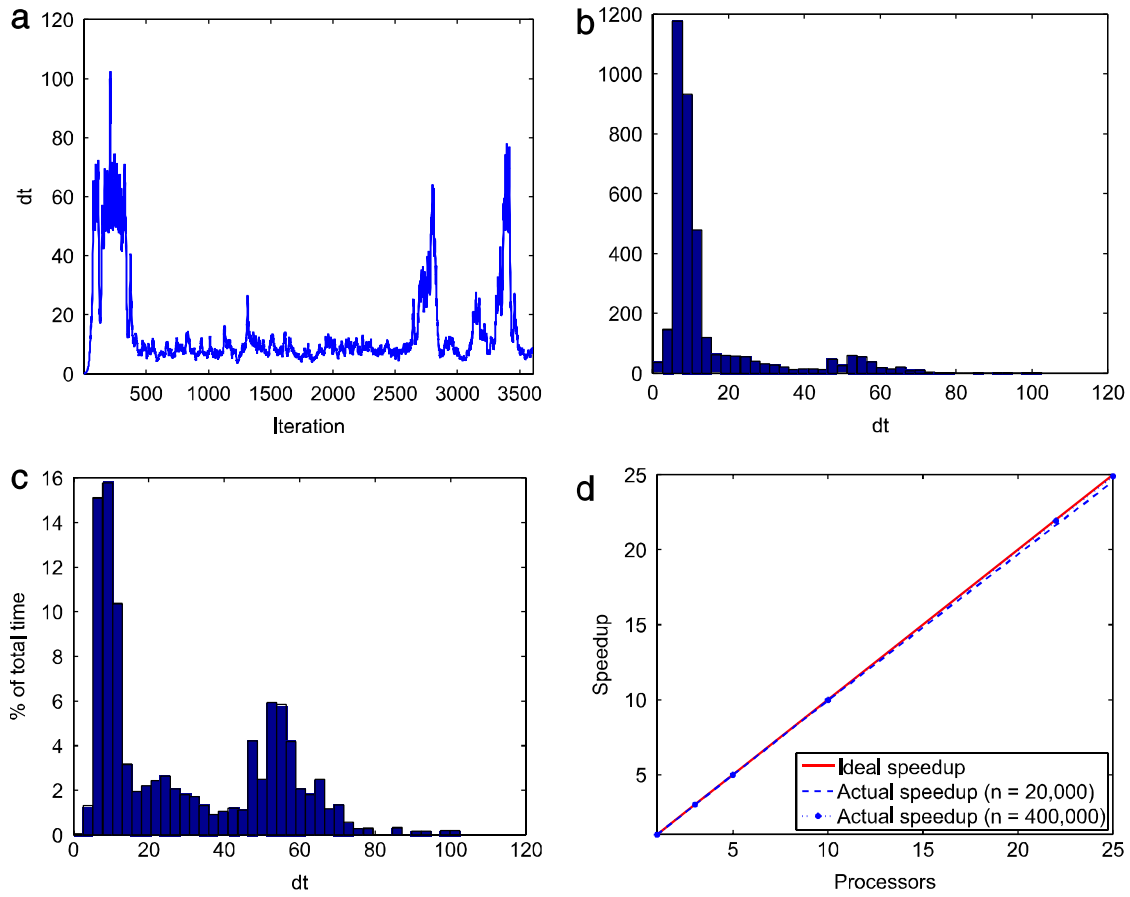


Fig. 6. Simulation results showing (a) value of dt as a function of iteration for a single track, (b) distribution of dt, (c) the percentage of total time covered by different ranges of those dt values, and (d) ideal and actual speed up for simulations using 20,000 and 400,000 particles respectively.

The techniques used in this paper can also be used for contaminant dispersion in e.g., groundwater pollution, as well as for forward and reverse particle transport models used e.g., in predicting atmospheric pollution dispersion. In future models we would like to incorporate full dispersion tensors, without the need for zero off-diagonal entries, and allow the use of space varying diffusion coefficients.

Acknowledgements

We would like to thank the referees for their helpful comments and suggestions. We thank the RIKZ in the Netherlands for providing us with the data.

References

[1] H.B. Fischer, E.J. List, R.C.Y. Koh, J. Imberger, N.H. Brooks, Mixing in Inland and Coastal Waters, Academic Press, New York, 1979.
[2] J.R. Manson, S.G. Wallis, An Accurate Numerical Algorithm for Advective Transport. Communication in Numerical Methods in Engineering, vol. 11, John Wiley and Sons, Ltd, 1995, pp. 1039–1045.
[3] G.J.M. Uffink, Analysis of dispersion by the random walk method, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1990.
[4] R.W. Barber, N.P. Volakos, Modelling depth-integrated pollution dispersion in the gulf of Thermaikos using a Lagrangian particle technique, in: M. de Conceicao Cunha (Ed.), Proc. of the 3rd of Inter. Confer. on Water Resources Management, Portugal, UK, in: WIT Trans. on Ecol. and the Env., vol. 80, 2005, pp. 173–184.
[5] A.W. Heemink, Stochastic modeling of dispersion in shallow water, Stoch. Hydrol. Hydraul. 4 (1990) 161–174.
[6] J.W. Stijnen, A.W. Heemink, H.X. Lin, An efficient 3d particle transport model for use in stratified flow, Internat. J. Numer. Methods Fluids 51 (3) (2006) 331–350.
[7] J.G. Gaines, T.J. Lyons, Variable stepsize control in the numerical solutions of SDEs, SIAM J. Appl. Math 57 (5) (1997) 1455–1484.
[8] P.M. Burrage, K. Burrage, Variable stepsize implementation for stochastic differential equations, SIAM J. Appl. Math. 24 (3) (2002) 848–864.
[9] W.M. Charles, E. van den Berg, H.X. Lin, A.W. Heemink, Variable time stepping in parallel particle models for transport problems in shallow waters, in: C.A. Brebbia, J.S.A. do Carmo (Eds.), Proc. of the 8th International Conference on Modelling, Monitoring and Management of Water Pollution, in: WIT Transactions on Ecology and the Environment, 2006, pp. 465–475.
[10] R.A. Trompert, J.G. Verwer, Analysis of the implicit Euler local uniform grid refinement method, SIAM J. Sci. Comput. 14 (1993) 259–278.
[11] P.E. Kloeden, E. Platen, H. Schurz, Numerical Solutions of SDE through Computer Experiments, Springer, New York, 2003.
[12] P.S. Addison, Bo Qu, A. Nisbet, G. Pender, A non-Fickian particle-tracking diffusion model based on fractional Brownian motion, Internat. J. Numer. Methods Fluids 25 (0) (1997) 1373–1384.
[13] A.W. Heemink, Tidally averaged models for dispersion in shallow water, Stoch. Hydrol. Hydraul. 3 (1993) 607–617.

[14] J.W. Stijnen, A.W. Heemink, Numerical treatment of stochastic river quality models driven by colored noise, Water Resour. Res. 39 (3) (2003) 1–9.
[15] L. Arnold, Stochastic Differential Equations: Theory and Applications, Wiley, London, 1974.
[16] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
[17] B. Oksendal, Stochastic Differential Equations. An Introduction with Applications, Springer, New York, 2003.
[18] E. van den Berg, A.W. Heemink, H.X. Lin, J.G.M. Schoenmakers, Probability density estimation in stochastic environmental models using reverse representations, Serra 20 (1–2) (2006) 126–139.
[19] G.S. Stelling, Communications on construction of computational methods for shallow water flow problems, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1983.
[20] D. Spivakovskaya, A.W. Heemink, E. Deleersnijder, Lagrangian modelling of multi-dimensional advection-diffusion with space-varying diffusivities: Theory and idealized test cases, Ocean Dyn. 57 (3) (2007) 189–203.
[21] P.E. Kloeden, E. Platen, Numerical Solutions of Stochastic Differential Equations. Application of Mathematics, Stochastic Modelling and Applied Probability, Springer-Verlag, Berlin Heidelberg, 1999.
[22] H.X. Lin, A.W. Heemink, J.W. Stijnen, Parallel simulation of the transport phenomena with the particle model Simpar, in: B.H. Li (Ed.), Proc. of the 4th Inter. Confer. on System Simulation and Scientific Computing, Academic Press, Beijing, 1999, pp. 17–22.
[23] The distributed ASCI supercomputer home page, www.cs.vu.nl/das2, 2005.