Variable time stepping in parallel particle models for transport problems in shallow waters

W. M. Charles[1], E. van den Berg[2], H. X. Lin[1] & A. W. Heemink[1]
[1]*Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Department of Applied Mathematical Analysis, Delft, The Netherlands*
[2]*University of British Columbia, Canada*

Abstract

Stochastic differential equations (SDEs) are stochastic in nature. The SDEs under consideration are often called particle models (PMs). PMs in this article model the simulation of transport of pollutants in shallow waters. The main focus is the derivation and efficient implementation of an adaptive scheme for numerical integration of the SDEs in this article. The error determination at each integration time step near the boundary where the diffusion is dominant is done by a pair of numerical schemes with strong order 1 of convergence and that of strong order 1.5. When the deterministic is dominant we use the aforementioned order 1 scheme and another scheme of strong order 2. An optimal stepsize for a given error tolerance is estimated. Moreover, the algorithm is developed in such a way that it allows for a completely flexible change of the time stepsize while guaranteeing correct Brownian paths. The software implementation uses the MPI library and allows for parallel processing. By making use of internal synchronisation points it allows for snapshots and particle counts to be made at given times, despite the inherent asynchronicity of the particles with regard to time.

Keywords: adaptive schemes, Wiener processes, SDEs, particle model, variable stepsize, parallel computing, speed up.

WIT Transactions on Ecology and the Environment, Vol 95, © 2006 WIT Press
www.witpress.com, ISSN 1743-3541 (on-line)
doi:10.2495/WP060461

1 Introduction to PMs in shallow water transport problems

Coastal ecosystems may experience environmental threats due to for example oil spills that may come from tanker accidents, or toxic chemical from the establishment of industries along the coastal areas. These processes require a contingency management of the transport materials in shallow waters. Numerical simulation of SDEs is widely applied in modern scientific investigations (Kloeden *et al* [1]). However, accurate solutions are not always guaranteed, thus there is a constant need to improve the numerical approaches in the mathematical models.

Fixed stepsize implementations of numerical methods in traditional particle models have limitations. Moreover, the use of fixed small stepsizes in the numerical approximation of SDEs may become unnecessary in case the error is very small and large time steps suffice. In the simulation of pollutant transport in shallow waters using SDEs, smaller stepsizes are needed to stably integrate in highly irregular areas and vice versa. In such situations, it is advantageous to employ an adaptive scheme in the particle model. Gaines *et al* [2] and Burrage *et al* [3] introduced a variable timestepping procedure for the pathwise (strong) numerical integration of a system of SDEs.

The concept of adaptive schemes by mesh refining in Eulerian methods have been used in ([4]). Particle models do not suffer from numerical diffusion in the source points (Heemink [5], Barber *et al* [6]). However, even when using an adaptive scheme, the computational cost may become high due to small stepsizes or the large number of particles (Kloeden *et al* [1]). Fortunately particles are independent from one another, thus allow efficient use of parallel processing.

In this article we implement parallel program to speed up the computation. This article is organised as follows. The governing set of SDEs and their schemes are discussed in section 2. The procedure of determining the variable stepsizes is described in section 2.4. The adaptive parallel time stepping implementation is described in sections 3. The results appear in section 3.2. The concluding remarks are given in Section 4.

2 Adaptive strong approximation of SDE in modeling of pollution transport in shallow Waters

The displacement of pollutants in shallow waters is described by:

$$\begin{aligned} dX_t &\overset{\text{Itô}}{=} \left[U + \tfrac{D_{XX}(x,y)}{H}\left(\tfrac{\partial H}{\partial x}\right) + \tfrac{\partial D_{XX}(x,y)}{\partial x}\right] dt + \sqrt{2D_{XX}(x,y)}\,dW_n^x \\ dY_t &\overset{\text{Itô}}{=} \left[V + \tfrac{D_{YY}(x,y)}{H}\left(\tfrac{\partial H}{\partial y}\right) + \tfrac{\partial D_{YY}(x,y)}{\partial y}\right] dt + \sqrt{2D_{YY}(x,y)}\,dW_n^y \end{aligned} \qquad (1)$$

(X_t, Y_t) is the position of a particle, $(U, V)^{\mathrm{T}}$ is flow velocities and H is the total water depth. Wiener processes $W_n^x(t)$ and $W_n^y(t)$ are Gaussian (Kloeden *et al* [1]), $D_{XX}(x, y)$ and $D_{YY}(x, y)$ are the horizontal dispersion coefficient functions in the x and y direction respectively.

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

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

where $D_{1,1}$ and $D_{2,2}$ are the horizontal dispersion parameters.

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

with $\underline{e}_k$ the k^{th} column from the identity matrix, $\underline{c} = (c_1, c_2)^{\mathrm{T}}$. Finally, α is the assumed to be the angle between the boundary and x or y direction. Where $\underline{c}$ is a direction vector a long the side of a given boundary cell, (xb, yb) is intersection point on the boundary between the line from (x, y) perpendicular to the boundary. $K \geq 0$ is a parameter modeling the decrease of diffusion coefficient near the boundary.

In numerical methods, there are two ways of measuring accuracy, namely strong convergence and weak convergence (Kloeden *et al* [1]). In this paper we make use of the strong convergence in determining the error at each step.

Definition 1 *Strong order of Convergence*
Let $\bar{X}_N$ be the numerical approximation of $X(T)$ after N steps. Under suitable conditions of the SDEs, for a fixed time T, the strong order of convergence is β_1 if there exist a positive constant $\mathcal{K}$ independent of Δt where $T = N\Delta t$, so the global order is defined:

$$\mathbb{E}\{|\bar{X}_N - X(T)|\} \leq \mathcal{K}(\Delta t)^{\beta_1}, \quad \mathbb{E}\{|\bar{Y}_N - Y(T)|\} \leq \mathcal{K}(\Delta t)^{\beta_1},$$

In this article we use pairs of schemes having different orders of convergence, in this way an error can be cheaply estimated at each step. Integration of the stochastic integral can be done using either the Itô or Stratonovich rule (Kloeden *et al* [1]). In this article, the Stratonovich rule is used in section 2.3, otherwise we use the Itô rule.

2.1 A scheme with strong order 1

Here we only give a brief overview of schemes, an interested reader is referred to Kloeden *et al* [1]. Consider the following scheme:

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

The expression for Y_{n+1} is similar to the above equation, with the first r.h.s. term X_n replaced by Y_n, all $D_{XX}(\cdot,\cdot)$ terms by $D_{YY}(\cdot,\cdot)$, the superscripts x modified to y. Where $\Delta W_n^x = W^x(t_{n+1}) - W^x(t_n)$ is an independent increment of Wiener processes in the time interval $[t_n, t_{n+1}]$. $n = 0, 1, \cdots$.

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

Similarly for Y_{n+1}^{*+1} along y direction. A drift function a^1 is given below:

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

likewise for a^2 along y direction.

2.2 A scheme with strong order 1.5

The following scheme (Kloeden *et al* [1]), is implemented in this article.

$$\begin{aligned} X_{n+1} \overset{\text{Itô}}{=} X_n &+ \left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) - a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} \times \frac{\Delta t_n}{4}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right) \\ &+ \frac{\Delta t_n}{4}\left\{a_n^{1+}(X_{n+1}^{*+z}, Y_{n+1}^{*+z}) + a_n^{1-}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})\right\} + \sqrt{2D_{XX}(X_n, Y_n)}\Delta W_n^x \\ &+ \frac{1}{4\sqrt{\Delta t}}\left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\}(\Delta W_n^x)^2 - \Delta t_n \\ &+ \left\{\sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})} - 2\sqrt{2D_{XX}(X_n, Y_n)} + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right\} \times \\ &\quad \left\{\Delta W_n^x - \frac{1}{2}\left(R_{n,1}^x + \frac{1}{\sqrt{3}}R_{n,2}^x\right)\sqrt{\Delta t_n}\right\} \\ &+ \left[\sqrt{2D_{XX}(X_{n+1}^{*+\phi}, Y_{n+1}^{*+\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*-\phi}, Y_{n+1}^{*-\phi})} - \sqrt{2D_{XX}(X_{n+1}^{*+z}, Y_{n+1}^{*+z})}\right. \\ &\quad \left. + \sqrt{2D_{XX}(X_{n+1}^{*-z}, Y_{n+1}^{*-z})}\right] \times \frac{1}{4\Delta t}\left\{\frac{1}{3}(\Delta W_n^x)^2 - \Delta t_n\right\}\Delta W_n^x. \end{aligned} \tag{6}$$

The expression for Y_{n+1} is similar to the above equation, with the first r.h.s. term X_n replaced by Y_n, all $D_{XX}(\cdot,\cdot)$ terms by $D_{YY}(\cdot,\cdot)$, the superscripts x modified to y for W and R, and similarly the $1+$ and $1-$ superscripts for a to $2+$ and $2-$. Using the shorthand notation of $\oplus$ for either $+$ or $-$, the following supporting vectors (used in equation 6) are defined

$$X^{*\oplus z}_{n+1} = X_n + \frac{1}{2} a^1_n (X_n, Y_n) \Delta t_n \oplus \sqrt{2D_{XX}(X_n, Y_n)\Delta t_n}$$

$$X^{*\oplus \phi}_{n+1} = X^{*\oplus z}_{n+1} \oplus \sqrt{2D_{XX}(X^{*+z}_{n+1}, Y^{*+z}_{n+1})\Delta t_n}.$$

The expressions for Y^{*+z}_{n+1}, Y^{*-z}_{n+1}, $Y^{*+\phi}_{n+1}$, and $Y^{*-\phi}_{n+1}$ are again similar, with the X in the first r.h.s. term replaced to Y, a^1_n replaced by a^2_n and D_{XX} by D_{YY}. consequently, using equations (5), we get

$$a^{1+}\left(X^{*+z}_{n+1}, Y^{*+z}_{n+1}\right) = \left[U + \frac{D_{XX}(X^{*+z}_{n+1}, Y^{*+z}_{n+1})}{H}\frac{\partial H}{\partial x} + \frac{\partial D_{XX}(X^{*+z}_{n+1}, Y^{*+z}_{n+1})}{\partial x}\right]$$

$$a^{1-}\left(X^{*-z}_{n+1}, Y^{*-z}_{n+1}\right) = \left[U + \frac{D_{XX}(X^{*-z}_{n+1}, Y^{*-z}_{n+1})}{H}\frac{\partial H}{\partial x} + \frac{\partial D_{XX}(X^{*-z}_{n+1}, Y^{*-z}_{n+1})}{\partial x}\right]$$

likewise for $a^{2+}\left(X^{*+z}_{n+1}, Y^{*+z}_{n+1}\right)$ and $a^{2-}\left(X^{*-z}_{n+1}, Y^{*-z}_{n+1}\right)$. $D_{XX}(\cdot,\cdot)$, $D_{YY}(\cdot,\cdot)$ approach zero toward the boundary and remain constant away from the boundary. Thus we are confronted with the situation where the drift becomes deterministic. The error criterion in this case holds for a pair of schemes of order 1 and higher strong order 2 of convergence, for example.

2.3 A scheme with strong order 2

Next we consider according to Kloeden *et al* [1], the following scheme:

$$\begin{aligned} X_{n+1} \overset{\text{Strat}}{=} X_n &+ \frac{1}{2}\left\{\underline{a}^1\left(X^+_{n+1}, Y^+_{n+1}\right) + \underline{a}^1\left(X^-_{n+1}, Y^-_{n+1}\right)\right\}\Delta t_n \\ &+ \frac{1}{\Delta t_n}\left\{\sqrt{2D_{XX}(t_n+1)} - \sqrt{2D_{XX}(t_n)}\right\}\left\{\Delta W^x_n \Delta t_n - \Delta M^x_n\right\} \\ &+ \sqrt{2D_{XX}(X_n, Y_n)}\Delta W^x_n \end{aligned} \tag{7}$$

$$\begin{aligned} Y_{n+1} \overset{\text{Strat}}{=} Y_n &+ \frac{1}{2}\left\{\underline{a}^2\left(X^+_{n+1}, Y^+_{n+1}\right) + \underline{a}^2\left(X^-_{n+1}, Y^-_{n+1}\right)\right\}\Delta t_n \\ &+ \frac{1}{\Delta t_n}\left\{\sqrt{2D_{YY}(t_n+1)} - \sqrt{2D_{YY}(t_n)}\right\} \times \left\{\Delta W^y_n \Delta t_n - \Delta M^y_n\right\} \\ &+ \sqrt{2D_{YY}(X_n, Y_n)}\Delta W^y_n. \end{aligned} \tag{8}$$

Here $D_{XX}(t) = D11$ and $D_{YY}(t) = D22$ are constants, so that the second line of the above two equations reduces to zero. The supporting vectors are defined by

$$\begin{aligned} X^{\oplus}_{n+1} &= X_n + \frac{1}{2}\underline{a}^1(X_n, Y_n)\Delta t_n \\ &\quad + \frac{1}{\Delta t_n}\sqrt{2D_{XX}(X_n, Y_n)}\left\{\Delta M^x_n \oplus \sqrt{2J^{x,p}_{(1,1,0)}\Delta t_n - (\Delta M^x_n)^2}\right\} \quad (9) \\ Y^{\oplus}_{n+1} &= Y_n + \frac{1}{2}\underline{a}^2(X_n, Y_n)\Delta t_n \\ &\quad + \frac{1}{\Delta t_n}\sqrt{2D_{YY}(X_n, Y_n)}\left\{\Delta M^y_n \oplus \sqrt{2J^{y,p}_{(1,1,0)}\Delta t_n - (\Delta M^y_n)^2}\right\}, \end{aligned}$$
$$(10)$$

where $\oplus$ the plus or minus operator. The definition of $\underline{a}^1(X, Y)$ is obtained by using Itô-Stratonovich transformation (see Kloeden *et al* [1]) of Eqn (5), yielding

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

likewise for $\underline{a}^2(X, Y)$ along y direction. Higher order schemes such as that of order 2, require the approximation of multiple higher Stratonovich stochastic integrals ($J^p_{(1,1,0)}$, see equation 11).

However, these cannot always be expressed in terms of simpler stochastic integrals, especially when the Wiener process is multi-dimensional. Using a method for multiple Stratonovich based on Kahunen-Loève or random Fourier series expansion of the Wiener process (for details, see Kloeden *et al* [1]) we can nevertheless approximate the integrals. This introduces a Brownian bridge into our model, a process fully described in Kloeden *et al* [1]. 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 $\left\{W_t - \frac{t}{T}W_T, 0 \leq t \leq T\right\}$. This can be done by generating an unconstrained (standard) Wiener process which is then linearly scaled in order to meet the required end points.

Following Karhunen-Loève (see [1]) we define the random variables a^x_r and b^x_r by

$$a^x_r = \frac{2}{\Delta t}\int_0^{\Delta t}\left(W^x_s - \frac{s}{\Delta t}W^x_{\Delta t}\right)\cos\left(\frac{2r\pi s}{\Delta t}\right)ds$$
$$\text{and} \quad b^x_r = \frac{2}{\Delta t}\int_0^{\Delta t}\left(W^x_s - \frac{s}{\Delta t}W^x_{\Delta t}\right)\sin\left(\frac{2r\pi s}{\Delta t}\right)ds, \quad r = 1, 2, \ldots$$

and likewise a^y_r and b^y_r, 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 $r \geq 1$ these variables have an $\mathcal{N}\left[0, \frac{\Delta t}{2\pi^2 r^2}\right]$ distribution. They are differentiable samples paths on the interval $[0, T]$.

Let $\zeta_r^x, \xi^x, \zeta_r^y, \xi^y, \eta_r^x, \eta_r^y, \phi_p^x,$ and ϕ_p^x denote independent random variables (Kloeden *et al* [1]), for $r = 1, 2, \ldots$ and $p = 1, 2, \ldots$:

$$\begin{array}{lll} \xi^x = \frac{1}{\sqrt{\Delta t}} W_{\Delta t}^x & \zeta_r^x = \sqrt{\frac{2}{\Delta t}}\pi r a_r^x & \eta_r^x = \sqrt{\frac{2}{\Delta t}}\pi r b_r^x \\ \mu_p^x = \frac{1}{\sqrt{\Delta t \rho_p}} \sum_{r=p+1}^{\infty} a_r^x & & \phi_p^x = \frac{1}{\sqrt{\Delta t \beta_p}} \sum_{r=p+1}^{\infty} \frac{1}{r} b_r^x \\ \mu_p^y = \frac{1}{\sqrt{\Delta t \rho_p}} \sum_{r=p+1}^{\infty} a_r^y & & \phi_p^y = \frac{1}{\sqrt{\Delta t \beta_p}} \sum_{r=p+1}^{\infty} \frac{1}{r} b_r^y. \end{array}$$

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 $\mathrm{var}[a_r^x] = \Delta t/2\pi^2 r^2$ (see Kloeden *et al* [1]) and with the fact that $\sum_{r=1}^{\infty} 1/r^2 = \pi^2/6$ and $\sum_{r=1}^{\infty} 1/r^4 = \pi^4/90$.

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

using the definition of a_r^x, a_r^y, and for each component and $r = 1, \ldots, p$ with $p = 1, 2, \ldots$, where p is the truncation index in the approximation of multiple integrals. We then define

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

Furthermore, we have

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

and similar for ΔM_n^y and $C_{y,y}^p$ and with superscripts changed from x to y. Using these random variables it turns out after lengthy computations that we can approximate a multiple integral as follows

$$\begin{array}{rcl} J_{(1,1,0)}^{x,p} & = & \frac{1}{6}(\Delta t)^2(\xi^x)^2 + \frac{1}{4}\Delta t(a_0^x)^2 - \frac{1}{2\pi}(\Delta t)^{\frac{3}{2}}\xi^x B^x \\ & + & \frac{1}{4}(\Delta t)^{\frac{3}{2}} a_0^x \xi^x - (\Delta t)^2 C_{x,x}^p. \end{array} \tag{11}$$

$J_{(1,1,0)}^{x,p}$ is an approximation of $J_{(1,1,0)}^x$ and it is known [1] 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 Eqns (7)-(8).

2.4 Determination of variable time stepsizes

Let $(\hat{X}_{n+1}, \hat{Y}_{n+1})$ be the numerical result obtained from the approximations of an SDE (1) using scheme (4) again we apply scheme (6) on the same particle near the boundary But when the drift term is dominant i.e., away from the boundary, we use Eqns (4) and scheme (7)- (8) where $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$ is due to a reference a higher scheme. $(X_{\mathrm{ref}_{n+1}}, Y_{\mathrm{ref}_{n+1}})$ is used to advance the numerical computation in the next time step, while $(\hat{X}_{n+1}, \hat{Y}_{n+1})$ and $(X_{\text{ref}_{n+1}}, Y_{\text{ref}_{n+1}})$ is used to estimate absolute error ([3]). Let tol_i be the tolerance accepted for the ith components then an error estimate of order $q + \frac{1}{2}$ in two-dimensional adaptive particle model:

$$error = \sqrt{\frac{1}{2}\left(\left|\frac{X_{\text{ref}_{n+1,1}} - \hat{X}_{1n+1,1}}{tol_1}\right| + \left|\frac{Y_{\text{ref}_{n+1,2}} - \hat{Y}_{1n+1,2}}{tol_2}\right|\right)}, \qquad (12)$$

where q is considered to be either $\hat{o}$ or o. Burrage *et al* [3] interpreted the calculated error as an approximation to the error in the higher order method unlike in the deterministic construction of ODEs. It is desirable that $X_{\text{ref}_{n+1,1}} - \hat{X}_{n+1,1} \approx tol_1$ and $Y_{\text{ref}_{n+1,2}} - \hat{Y}_{n+1,2} \approx tol_2$, the step just completed is rejected if $error > 1$ otherwise compute an optimal stepsize $(\Delta t)_{opt} = \Delta t_{old}\left(\frac{1}{error}\right)^{\frac{1}{2}}$ until the desired accuracy is attained. For efficient implementation using a variable stepsize strategy, an optimal stepsize can be decreased by a safety factor for example 0.8 to avoid oscillatory behaviour in the stepsize so that it does not increase or decrease too quickly [3]:

$$(\Delta t)_{new} = \Delta t_{old} * \min\left(facmx, \max\left(facmn, fac * \left(\frac{1}{error}\right)^{\frac{1}{2}}\right)\right) \qquad (13)$$

where $facmx$ and $facmn$ are the maximal and minimal stepsize scaling factors allowed, respectively for the problems being solved (Burrage *et al* [3]). Variable stepsize implementation has a possibility of stepsize acceleration using Eqn. (13). This arises when a step fails, possibly due to extreme random sample, in this article, we avoid uncontrolled jumps in the step size such that the final step length is given by

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

3 Implementation of time stepping adaptive parallel processing for SDEs

The implementation of adaptive scheme differs substantially from one with a fixed step size in that is it no longer possible to have a single major loop governing the time by taking a single step of fixed size (Lin *et al* [7]). Instead, the current time differs between the particles and, in addition to the coordinates, each particle now needs a local time associated with. This concept of local time introduces a wide level of asynchronicity into the model, making it hard to define a major loop in the traditional way. Additionally, this lack of synchronous time complicates taking a snapshot of the particle locations at a given time.

To overcome these difficulties we introduce an event mechanism which defines certain synchronisation points in the otherwise chaotic time line. The implementation consists of a number of different modules, each taking care of a certain function within the program. Each module provides the central engine with a list of desired events consisting of the time(s) at which they should occur, a type and possible some additional data. It then invokes the integration module with the present time and the time to integrate to. The integration routine will then perform the integration and is completely free to decide how this time interval is integrated. It will ensure however that each particle is exactly integrated up to the desired ending time, coinciding with the event, unless of course the particle flows out of the domain before that. This way, the result of the integration call is a set of particles, with their location at exactly the time of the event. The main program itself also generates an event telling the main loop to stop at the desired time. The particle model lends itself extremely well for parallel processing, since the particles do not interact with one another and can therefore be considered on an individual basis. By dividing the particles, instead of the domain, across the processors we take full advantage of the independency.

3.1 Parallel processing experiments

Experiments of prediction of the dispersion of pollutants are carried out on a distributed memory parallel architecture called DAS-2 [8]. It is a 200-node system with a total of 400 -processors wide-area distributed system. *Speed up* is the ratio of the time taken to solve a problem on a single processor to the time required to solve the same problem on a parallel computer with *p* processors: $S(p) = \frac{T_1}{T_p}$, for speed up result see Fig.1(d).

3.1.1 Summary of the simulation parameters

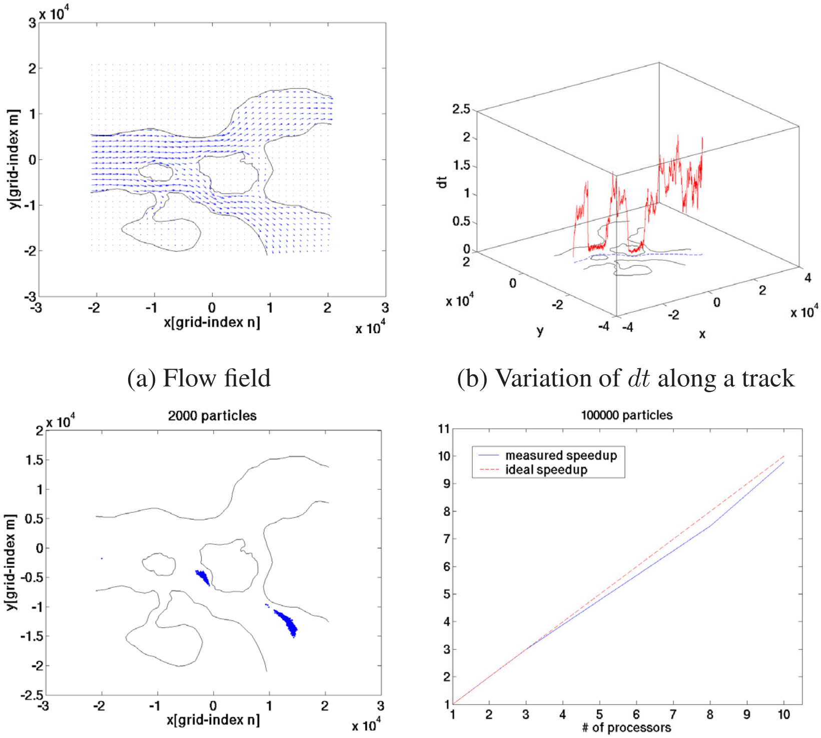
Grid size 105×105, $tol1 = tol2 = 12$, minimum Δt=$0.0001s$, initial $\Delta t = 0.1s$, $p = 10$, $D11 = D22 = 10m^2/s$, initial point $(-20000m, -1800m)$, $\Delta x = \Delta y = 400m$, $H(x, y) = 10m$, $fac = 0.8$, $facmin = 0.6$, $facmax = 1.1$, $K = 1m$. Radius =3, is the number of grid rings surrounding the threshold point. Threshold distance= $1000m$ is the point where the two schemes of order 1.5 and order 2 exchange, Brownian bridge steps= 30.

3.2 Results

The following results Fig. 1 (a)-(c) carried out by one processor using a domain composed of the river,the lake and two islands as well as outflows. 2000 particles were initially released at the point $(-20000m, -1800m)$.

4 Concluding remarks

In this paper an adaptive scheme for the parallel simulation of pollutant transport in shallow waters using SDEs has been implemented. We have seen that smaller stepsizes are needed to stably integrate in highly irregular areas and vice versa, see Fig. 1(b). Thus, it is advantageous to employ an adaptive scheme in the particle



(a) Flow field

(b) Variation of dt along a track

(c) Snap shot taken after 5 minutes

(d) Speedup

Figure 1: Simulation results (a) flow fields, (b) variations of stepsize at different locations, (c) snapshot of particles' position at every 5 minutes, (d) speed up measured on a Beowulf cluster.

model. Good speed up is attained as well. As a consequence, at least at the moment there is no need to carefully divide the domain into several sub-regions. But more analysis will be carried out.

Acknowledgements

The authors thank Dr. J.A.M. van der Weide for his helpful and stimulating discussions. This work is supported by TUDelft and UDSM.

References

[1] Kloeden, P.E., Platen, E. & Schurz, H., *Numerical solutions of SDE Through Computer Experiments*. Springer: New York, pp. 1–292, 2003.

[2] Gaines, J.G. & Lyons, T.J., Variable stepsize control in the numerical solutions

of stochastic differential equations. *SIAM J APPL MATH*, **57(5)**, pp. 1455–1484, 1997.

[3] Burrage, P.M. & Burrage, K., Variable stepsize implementation for stochastic differential equations. *SIAM J APPL MATH*, **24(3)**, pp. 848–864, 2002.

[4] Trompert, R.A. & Verwer, J.G., Analysis of the implicit Euler local uniform grid refinement method. *SIAM J Sci Comput*, **14**, pp. 259–278, 1993.

[5] Heemink, A.W., Stochastic modeling of dispersion in shallow water. *Stochastic Hydrology and hydraulics*, **0(4)**, pp. 161–174, 1990.

[6] Barber, R. & Volakos, N., Modelling depth-integrated pollution dispersion in the gulf of thermaikos using a lagrangian particle technique. *Proc. of the 3rd of Inter. Confer. on Water Resources Management, Portugal*, ed. M.D.C. Cunha, WIT Trans. on Ecol. and the Env.., Vol 80: UK, pp. 173–184, 2005.

[7] Lin, H.X., Heemink, A.W. & Stijnen, J.W., Parallel simulation of the transport phenomena with the particle model Simpar. *Proc. of the 4th Inter. Confer. on SYSTEM SIMULATION AND SCIENTIFIC COMPUTING*, ed. B.H. Li, Distributed inter.Academic press: Beijing, pp. 17–22, 1999.

[8] The distributed ASCI supercomputer home page. *www cs vu nl/das2*.