JOURNAL OF APPLIED COMPUTER SCIENCE & MATHEMATICS, ISSUE 1 / 2025, VOL. 19, NO. 38
ISSN: 2066-4273 eISSN: 2066-3129

Analysis of the efficiency and accuracy of two adaptive partitioned time step schemes in Lagrangian particle tracking algorithms

[1]Valdir Innocentini, [2]Luciana de Freitas Tessarolo, [3]Julio Tomás Aquije Chacaltana , [4]Ernesto Caetano, [5]Fernando Túlio Camilo Barreto, [6]Audálio Rebelo Torres Júnior, [7]Tânia Ocimoto Oda, [6]Júlio César Martins Ribeiro Júnior

[1]Instituto Nacional de Pesquisas Espaciais (INPE), Brazil,
[2]Rede de Modelagem e Observação Oceanográfica (REMO), Centro de Hidrografia da Marinha (CHM), Brazil,
[3]Laboratório de Simulação de Escoamentos com Superfície Livre (LABESUL), Universidade Federal do Espírito Santo (UFES), Brazil,
[4]Instituto de Geografía, Universidad Nacional Autónoma de México (UNAM), Mexico
[5]OceanPact, Brazil,
[6]Laboratório de Hidrodinâmica Costeira, Estuarina e de Águas Interiores (LhiCEAI), Universidade Federal do Maranhão (UFMA), Brazil,
[7] Instituto de Estudos do Mar Almirante Paulo Moreira (IEAPM), Brazil.
[1]vinnocentini@hotmail.com, [2]lutessarolo@hotmail.com

Abstract
Simulations of particle trajectories on the ocean's surface usually involve thousands of releases. The efficiency and accuracy of Lagrangian trajectory algorithms, based on the velocities provided at the nodes of an Eulerian grid, depend mainly on the size of the time step (Δt) to calculate the movement between two consecutive points that make up the trajectory. Adaptive partitioned time step schemes are viable alternatives, as the Δt can be divided into parts to increase accuracy when the speed is high, but, to maintain efficiency, preserve the original Δt when the speed is low. This study proposed two algorithms with adaptive particle tracking formulations: in one of them, the particle cannot cross the edge of the grid cell in a single movement, and in the other, it can, but to a limited extent. They were compared in a hydrodynamic simulation of currents forced by winds and tides in a bay with channels, islands and obstacles, with large variations in grid cell size and surface velocity. The two algorithms showed similar accuracy and efficiency, with some differences in channels where velocity variations were greater. The size of the Δt had a notable impact on computing time; the choice of the size should be decided taking into account the desired details for trajectories. In cases of continuous leakage, Δt cannot be greater than the time interval between two consecutive releases.

Keywords: *Adaptive partitioned time step, Lagrangian trajectory algorithms, ill-conditioned intersection situations.*

I. INTRODUCTION

Lagrangian Particle Models (LPMs) for simulating particle trajectories in aquatic environments are crucial for providing information to emergency personnel in cases of oil spills and search and rescue operations. In its most basic form, LPM uses the velocity obtained from hydrodynamic models to calculate the position after a displacement in a time interval Δ*t*. Many displacements make up the complete trajectory for the total desired period. The integral equation involving the velocity is discretized, and a new position $\mathbf{P}_F$ is computed by some numerical method after the time step Δ*t*:

$$\mathbf{P}_F = \mathbf{P}_I + \int_{t_I}^{t_{I+\Delta t}} \mathbf{U}\, dt \tag{1}$$

Using the Runge-Kutta formalism, for the second-order approach (RK-2) the new position is given by [1]

$$\mathbf{P}_F = \mathbf{P}_I + \sum_{i=1}^{2} a_i \mathbf{K}_i \tag{2}$$

where

$$\begin{aligned} \mathbf{K}_1 &= \Delta t\, \mathbf{U}\left(t_I, \mathbf{P}_I\right) \\ \mathbf{K}_2 &= \Delta t\, \mathbf{U}\left(\mathrm{t}_I + \Delta t, \mathbf{P}_I + \mathbf{K}_1\right) \\ a_1 &= a_2 = 0.5 \end{aligned} \tag{3}$$

In practical applications the trajectory of just one particle is not enough, as small variations in the initial position and time can produce quite different results [2]. The usual is to obtain thousands of trajectories and analyze the results through statistical interpretations of particle concentrations. Overall, LPM simulations must have accuracy and efficiency. However, while a higher Δ*t* increases efficiency, precision is compromised. To combine both requirements, adaptative time step scheme (ATS) is the ideal solution, where Δ*t* is small when (or where) the speed is large, but when the speed becomes small Δ*t* can be increased.

In this study, two algorithms are proposed for LPM with ATS (section II), contrasting in defining the limits of small displacements. Appropriate tools are needed for each approach. Alternative formulations to the tools are compared and the most efficient are implemented in ATM algorithms,

DOI: 10.4316/JACSM.202501001

ISSN: 2066-4273 eISSN: 2066-3129

which are tested in a simulation of a hypothetical oil spill in a bay where the particles are transported by currents provided by the Delft3D hydrodynamic model [3], due to the tides and wind (section III). The results and discussion are presented in section IV and, finally, the main conclusions are summarized in section V.

II. Material And Methods

In general, LPM is applied using off-line model outputs with velocities supplied on the grid nodes at discrete times, which are separated by a period that depends on the spatial scale of the simulation. In this study, the period was 1 h.

For each displacement, the algorithm is made up of two parts: movement and location. At each time interval, the particle moves to a new position due to advection, then the cell containing the particle needs to be known, so that the velocity at that position is found and a new movement can be carried out. The velocity is obtained by interpolation using velocities at the cell nodes. In ATS, the velocity at the particle's position is renewed in several smaller time intervals δt, until the accumulated sum completes the basic time interval Δt adopted by the user, i.e.

$$\mathbf{P}_F^i = \mathbf{P}_F^{(i-1)} + \int_{t_{(i-1)}}^{t_{(i-1)}+\delta t_i} \mathbf{U}\, dt, i = 1,...,n \qquad (4)$$

where $\mathbf{P}^0_F = \mathbf{P}_I$ is the initial position, and $\mathbf{P}_F = \mathbf{P}^n_F$ is the final position at $t_I + \Delta t$, with $t_i = t_{(i-1)} + \delta t_i$, being δt_i the time necessary to move the particle from $\mathbf{P}^{(i-1)}_F$ to $\mathbf{P}^i_F$.

Figure 1 illustrates the two ATS algorithms proposed here, EdInter and EdCross, explained below.

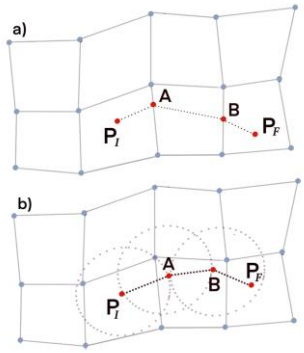

Fig. 1 Illustrative examples of ATS algorithms. a) EdInter: the particle starting at $\mathbf{P}_I$ is intercepted at **A** and **B**, before consuming the full time Δt. b) EdCross: each movement of the particle has a maximum distance defined by the starting cell. Points **A** and **B** are at their maximum, but $\mathbf{P}_F$ falls short, limited by the time remaining to complete Δt.

A) EdInter algorithm

In this algorithm, each small part of the time step is computed so that the particle is intercepted at the edge of the cell. Then a new velocity is determined and a new movement is carried out. The process continues until the time step Δt has expired. In the example illustrated in Figure 1.a it was necessary to divide Δt into 3 δts producing the segments: **($\mathbf{P}_I$ A)**, **(A B)**, and **(B $\mathbf{P}_F$)**. The sum of the times δts needed to make each movement is equal to Δt. Note that if the velocity is too low, **($\mathbf{P}_I$ $\mathbf{P}_F$)** is inside the cell and there is no intersection. EdInter requires testing which edge is crossed by the trajectory and obtaining the intersection point.

B) EdCross algorithm

This algorithm allows the trajectory in a δt to go beyond the edge of the current cell to some neighboring cell, so that a predefined distance is not exceeded. This distance is valid only for each cell and must ensure that a movement never reaches a cell beyond the neighboring cells; it is the minimum length of the segments joining any point within the cell to any other in the neighboring cells. In the example illustrated in Figure 1.b, the dashed circles show the maximum distance the particle can reach. The length of **($\mathbf{P}_I$ A)** is the maximum distance for the path starting at $\mathbf{P}_I$ and, likewise, point **B** is the maximum distance from **A**, and $\mathbf{P}_F$ is so that the distance from **B** to $\mathbf{P}_F$ is less than the maximum distance allowed from **B**. EdCross requires testing which neighboring cell contains the end point of each movement.

C) Preprocessing informations

Both algorithms require information, which must be readily available, to be retrieved from preprocessed files. Assuming that each cell node is assigned an identification number ID, the Table 1 below summarizes these files and their properties.

The identification number ID of a cell is due to the premise that each node originates a cell. The coefficients in the files Vector_Around and Normal_Around are a, b, and c in the equation of the line $ax + by + c = 0$ which overlaps the edges and vectors mentioned.

Table 1. Description of preprocessing files

File	Description	Used by
Node_Location	Node coordinates for each ID node	both
Cell_Nodes	ID nodes of the cell	both
Vector_Around	Coefficients of the edge segments around the cell	both
Normal_Around	Coefficients of normal vectors to the edges	EdInter
Face_Cell	ID cell facing each edge of a cell	both
Cell_Dmax	Maximum displacement for a particle in the cell	EdCross

D) In-cell tests

Both algorithms need to test whether or not a position is inside a cell. EdInter uses the in-cell test to find out if the trajectory has crossed any edge of the current cell before calculating the intersection point, while EdCross needs to

DOI: 10.4316/JACSM.202501001

ISSN: 2066-4273 eISSN: 2066-3129

know which cell contains the position. We have selected 3 methods for in-cell tests:

1) barycentric coordinates, given by [4],
2) areas comparison, used by [5], and
3) looping inside the cell, presented by [6].

The three methods were tested on cells in the shape of triangles, quadrilaterals, and pentagons; the looping scheme was the most efficient. In this method, the vertices of the polygons are ordered counterclockwise, and the cross product between the vector on each edge and the vector connecting a vertex with the position $\mathbf{P}_F$ must be positive or zero.

E) Intersection point

The efficiency of 3 formulations to obtain the point of intersection between two lines was studied:

1) similarity of triangles,
2) Gauss elimination method, and
3) Cramer's elimination method.

In elimination methods, a system of two equations with two unknown variables (the *x* and *y* coordinates of the intersection point) is solved. The triangle similarity method was proposed by [7], but here a new interpretation makes it easier to identify ill-conditioned situations. In Figure 2, the trajectory is the segment $(\mathbf{P}_I\mathbf{P}_F)$ that crosses the edge indicated by the vertices $\mathbf{V}_i$ and $\mathbf{V}_{(i+1)}$. The vectors **t**, **n**, and **a** are unitary, in the direction of $(\mathbf{P}_I\mathbf{P}_F)$, normal to the segment $(\mathbf{V}_i\mathbf{V}_{(i+1)})$, and in its direction, respectively.

The intersection point $\mathbf{P}_{Int}$ along the segment $(\mathbf{P}_I\mathbf{P}_F)$ can be written as $\mathbf{P}_{Int} = \mathbf{P}_I + \beta_{Int}\,\mathbf{t}$, where β_{Int} is the length of $(\mathbf{P}_I\mathbf{P}_{Int})$. Observing that the sides with lengths $|\mathbf{t}|$ and $|\mathbf{t}\cdot\mathbf{n}|$, are similar to the sides with lengths β_{Int} and $|\mathbf{P}_I\mathbf{V}_i|\cdot\mathbf{n}$, it follows that β_{Int} is given by the quotient of two scalar products

$$\beta_{Int} = \frac{(\mathbf{P}_I\mathbf{V}_i)\cdot\mathbf{n}}{\mathbf{t}\cdot\mathbf{n}} \tag{5}$$

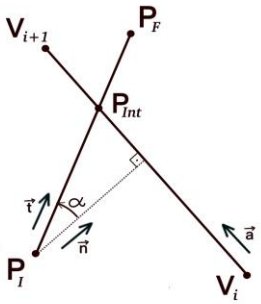

Figure 2. The $(\mathbf{P}_I\mathbf{P}_F)$ trajectory is crossing the edge of the cell with vertices $\mathbf{V}_i$ and $\mathbf{V}_{i+1}$, and $\mathbf{P}_{Int}$ is the intersection point. The vectors **t**, **a**, and **n** are unitary, along the trajectory, along and normal to the edge, respectively.

In situations where the lines are almost parallel, i.e. in ill-conditioned cases, the three methods can provide incorrect solutions. The triangle similarity method proved to be the most efficient, as well as making it possible to identify ill-conditioned cases when the angle α is close to 90°. We propose that if the trajectory is passing between two vertices of an edge, and the angle α with the edge is greater than 89°, the intersection point should be calculated by

$$\mathbf{P}_{Int} = \mathbf{V}_i + \gamma\mathbf{a} \tag{6}$$

where

$$\gamma = \min\left(|\mathbf{V}_i\mathbf{V}_{i+1}|, |\mathbf{V}_i\mathbf{P}_F|\right)$$

$$\mathbf{a} = \frac{(\mathbf{V}_{i+1} - \mathbf{V}_i)}{d_i} \tag{7}$$

$$d_i = |\mathbf{V}_i\mathbf{V}_{i+1}|$$

In this way $\mathbf{P}_{Int}$ is on the edge and between the vertices.

F) The ideal time step Δt

The proposed algorithms split the time step chosen by the user when necessary, but when the speed is very low, applying a multiple of Δt is not possible. Therefore, an unnecessarily short time step will increase the time spent on the computer. In terms of computational cost, if at a given instant in a cell with length L the velocity is V, the time step $\Delta t = L/V$ is the ideal time step. For the hydrodynamic velocity field over the entire grid, the minimum time interval that provides the Courant number equal to or less than 1 is the ideal time step defined as Δt_{Id}.

III. Experiments And Results

The study by [8] simulated currents on the mesh with the Delft3D hydrodynamic model [2] forced by hourly outputs provided by ERA5 [9]. The 2 algorithms for the trajectory, EdInter and EdCross, with the first and second-order Runge-Kutta approaches (RK1 and RK2, respectively), were applied to a hypothetical leak. The particle transport is carried out by currents, which showed great variation in time and space due to winds and tides. The extension of the domain is 24.5 and 18.0 km in the East and North directions, respectively, as represented in Figure 3. It consists of 271×281 points with minimum longitude and latitude -40.381° W and -20.391° S, respectively (on the western and southern limits). The coastal contour is very complex and requires cells of various sizes; their diagonal length varies from 24.85 m to 366.23 m. An analysis of the currents in the grid cells revealed Δt_{Id} = 80 s. Based on this result, in this study we propose experiments with Δt = 60, 240, and 1,200 s.

For 48 hours, the algorithms tracked 100,000 particles, with 2,500 being launched every 3600 s, from hour 0 to hour 9, at small boxes area 0.03° × 0.03° around 4 points in the middle of the bay (see Figure 3). To ensure that they didn't follow

DOI: 10.4316/JACSM.202501001

ISSN: 2066-4273 eISSN: 2066-3129

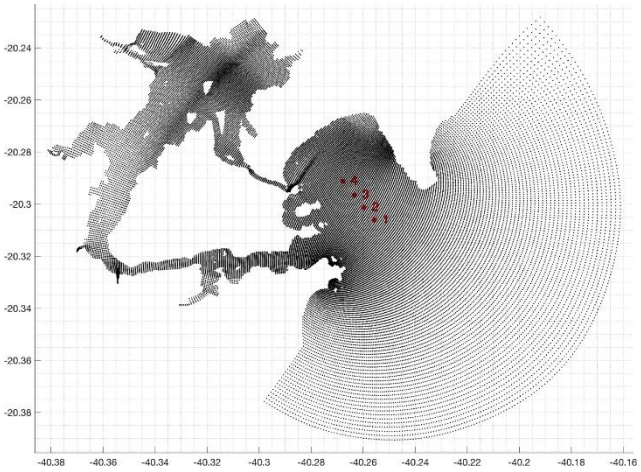

Figure 3. Curvilinear coordinates grid mesh for the Vitoria-Espírito Santo Bay System. Points 1,2,3, and 4 are locations where particles were released.

exactly the same path, a random walk was applied to the particles' displacements.

The CPU processor used was the Intel ® Xeon ® X5365@3.00GHz. A comparison of the time needed for the RK1 and RK2 methods shows that the latter spends around 40 to 50% more time. With regard to the effect of the size of the time step, the time taken by the two algorithms with the RK2 method is shown in Table 2: EdCross was faster than EdInter for small Δt (60 and 240 s), but its advantage decreased slightly with large Δt (1200 s). Both algorithms for Δt = 1200 s spent about 20-25% of the time spent for Δt = 60 s.

The particle trajectories showed some differences, which was to be expected, since in each algorithm and differents Δt the velocities used to compute the displacements are not exactly at the same locations. In the channels, where the velocities are higher and where there is a sudden change in direction and speed due to the tides, the differences are more noticeable.

For the probability analysis, the domain was divided into compartments of 0.03° × 0.03°, and the occurrence of some properties was counted in histograms. For example, the occupancy probability, which is the probability of a box being occupied by some particle during the entire integration period, cited by [10]. At the end of the 48 hours, the probability fields for the experiments carried out showed practically identical results, with some differences within the channels, where the grids were smaller and the speeds were higher and with sudden reversals of direction due to the tides.

TABLE 2. COMPUTER TIME SPENT BY ALGORITHMS FOR A PERIOD OF 48 H, FOR Δt =60, 240, AND 1200 S. THE TIME SPENT IS NORMALIZED BY THAT OF EDINTER 60.

Δt (s)	EdInter	EdCross
60	1	0.89
240	0.36	0.33
1200	0.18	0.21

The differences were probably masked or amortized due to the large number of particles released.

CONCLUSIONS

The choice of an algorithm for LPM depends crucially on the complexity of the currents provided by the hydrodynamical model and the region considered. For situations with large variations in the speed and direction of the currents, and the size of the grid cells, more sophisticated schemes to preserve efficiency and accuracy are required.

In principle, choosing a small time step to calculate two consecutive points on a trajectory is required for accuracy. However, efficiency is compromised in regions of the domain where the velocity is very small. Algorithms with ATS are the ideal choice for combining these two requirements.

Two algorithms were analyzed and applied to a situation with tides and wind forcing the currents in a region with bays and channels. In one method, the time step is divided into parts such that each point of the trajectory is the intersection of the trajectory with the side of a cell (EdInter), and in the other the point can reach a neighboring cell, but never more than a certain distance from the previous location (EdCross).

The choice of Δt should be based on the ideal time step, which is the one that provides a Courant number equal to or less than 1 for the velocity and grid cell length fields. In the case studied here, the ideal time step is close to 80 s. Simulations showed that the computational cost was reduced to 1/5 when comparing simulations with Δt = 60 s and 1200 s (for both algorithms). In the case studied, differences were observed in some trajectories obtained with different Δt for both algorithms, mainly in channels, where the speed and direction of the currents exhibited the greatest variations. However, comparisons between probability maps showed insignificant differences.

Although both ATS algorithms proposed here consume practically the same amount of computing time, we recommend using EdInter because the partitioning is more refined, i.e. the distance between 2 consecutive points is smaller, which implies greater precision without losing efficiency. As in this kind of simulation the release is only carried out in time steps, the value of Δt can be large in experiments where the leakage is not continuous.

Supplementary material

The numerical code for both algorithms, as well as their pseudocodes and a detailed explanation of their use, can be obtained by contacting the first author at vinnocentini@hotmail.com.

REFERENCES

[1]. Cheney, W., & Kincaid, D. (2007). Numerical Mathematics and Computing.Thomson Brooks/Cole, Belmont, CA.
[2]. Csanady, G.T. (2012). Turbulent Diffusion in the Environment. Springer Science & Business Media.

DOI: 10.4316/JACSM.202501001

ISSN: 2066-4273 eISSN: 2066-3129

[3]. Deltares, (2024). Delft3D: Functional Specifications. URL: https://content.oss.deltares.nl/delft3d4/Delft3D Functional_Specifications.pdf. Version: 2.20, Revision: 78359. Accessed: February 06, 2024.
[4]. Westermann, T. (1992). Localization schemes in 2D boundary-fitted grids. Journal of Computational Physics 101, 307–313. doi:https://doi.org/10.1016/0021-9991(92)90008-M.
[5]. Ogami, Y. (2021). Fast algorithms for particle searching and positioning by cell registration and area comparison. Trends in Computer Science and Information Technology 6, 007–016. doi:https://doi.org/10.17352/tcsit.000032.
[6]. Zhou, Q., & Leschziner, M. (1999). An improved particle-locating algorithm for Eulerian-Lagrangian computations of two-phase flows in general coordinates. International Journal of Multiphase Flow 25, 813–825. doi:https://doi.org/10.1016/S0301-9322(98)00045-7.
[7]. Haselbacher, A., Najjar, F., & Ferry, J. (2007). An efficient and robust particle-localization algorithm for unstructured grids. Journal of Computational Physics 225, 2198–2213. doi:https://doi.org/10.1016/j.jcp.2007.03.018.
[8]. Barreto, F.T.C. (2019). Modelling the fate and transport of oil spills. Ph.D. thesis. Universidade Federal do Espírito Santo (UFES). Vitória. URL: http://portais4.ufes.br/posgrad/teses/tese_13825_tese_fernando_tulio_camilo_barreto_compressed.pdf
[9]. ECMWF (2024). ERA5 data documentation. URL: https://confluence.ecmwf.int/display/CKB/ERA5 %3A+data+documentation. European Centre for Medium-range Weather Forecasts (ECMWF). Accessed: February06, 2024.
[10]. van Sebille, E., Griffies, S.M., Abernathey, R., Adams, T.P., Berloff, P., Biastoch, A., Blanke, B.,Chassignet, E.P., Cheng, Y., Cotter, C.J., Deleer-snijder, E., Döös, K., Drake, H.F., Drijfhout, S., Gary, S.F., Heemink, A.W., Kjellsson, J., Koszalka, I.M., Lange, M., Lique, C., MacGilchrist, G.A., Marsh, R., Mayorga Adame, C.G., McAdam, R., Nencioli, F., Paris, C.B., Piggott, M.D., Polton, J.A., Rühs, S., Shah, S.H., Thomas, M.D., Wang, J., Wolfram, P.J., Zanna, L., & Zika, J.D. (2018). Lagrangian ocean analysis: Fundamentals and practices. Ocean Modelling 121, 49–75. doi:https://doi.org/10.1016/j.ocemod.2017.11.008

Valdir INNOCENTINI is a mathematician with a PhD in meteorology at University of Reading. He is currently working on issues related to modelling oil spills and oil spreading at sea.

Luciana DE FREITAS TESSAROLO is a chemical engineer with a PhD in Meteorology. Her work focuses on numerical modeling used to simulate the behavior of oil and gas released in deepwater or on the surface.

Julio Tomás AQUIJE CHACALTANA is a hydrodynamicist. His research interests include circulation pattern recognition and contaminant fate and transport in aquatic environments. He is a full professor in the Environmental Engineering Department of the Federal University of Espírito Santo, Brazil.

Ernesto CAETANO has a PhD degree in Atmospheric Sciences from the University of Reading. He currently is researcher at National Autonomous University of Mexico, where he is involved in projects related to climate change, air quality, and wind energy potential.

Fernando Túlio CAMILO BARRETO is a physical oceanographer with a PhD in Environmental Engineering. He is a specialist in computational modeling, working in the private sector on metoceanographic and oil spill modeling.

Audálio REBELO TORRES JÚNIOR holds a PhD in Ocean Engineering. He is currently a professor at the Federal University of Maranhão. He has experience on numerical modeling, ocean circulation, and applied meteorology.

Tânia OCIMOTO ODA holds a D.Sc. degree in Atmospheric Sciences in Ocean Engineering. She is currently working as a researcher at Instituto de Estudos do Mar Almirante Paulo Moreira (Brazil). Her main interest are sea air interaction processes.

Júlio César MARTINS RIBEIRO JÚNIOR holds a master's degree in Oceanography. His current work focuses on the use, implementation, and development of numerical models to simulate ocean dynamics and environmental processes.

DOI: 10.4316/JACSM.202501001