HAL
open science

Simulation of drifting seaweeds in East China Sea

Jean Baptiste Filippi, T. Komatsu, K. Tanaka

To cite this version:

Jean Baptiste Filippi, T. Komatsu, K. Tanaka. Simulation of drifting seaweeds in East China Sea. Ecological Informatics, 2010, 5 (1), pp.67-72. 10.1016/j.ecoinf.2009.08.011 . hal-00433257

HAL Id: hal-00433257

https://hal.science/hal-00433257v1

Submitted on 16 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulation of drifting seaweeds in East China Sea

Jean-Baptiste Filippi[12]
filippi@univ-corse.fr

Teruhisa Komatsu[2]
komatsu@ori.u-tokyo.ac.jp

Kyushu Tanaka[2]
tanaka@ori.u-tokyo.ac.jp

[1] Corresponding Author

[2]Ocean Research Institute
The University of Tokyo
1-15-1 Minamidai, Nakano-ku
Tokyo 164-8639, JAPAN

ABSTRACT
Drifting seaweeds plays a major role in areas where they are present. We describe a computer model, Jeosim that clarifies the roles of those seaweeds in relation to their transport using particle tracking algorithm. The Euler rule with trapezoidal approximation, used to calculate drifting paths, is implemented in JeoSim in discrete events fashion in order to simulate the path and movement of seaweeds. Pre-processing of the ocean currents is done with the Princeton Ocean circulation Model (POM). Simulated current constitutes the data that provides force and directions for the calculations of drifting paths. Along with the implementation of the particle tracking algorithm in discrete event fashion, the originality of the JeoSim software lies in its Object Oriented architecture which makes it especially suited to perform simulation of living, cross scale systems with complex behavior. Behavior of drifting seaweeds of the *Sargasso* family has been implemented in JeoSim, and experimented in East China Sea following a drifting seaweed collection campaign in may 2002. Despite a relative low resolution ocean currents data, simulated results compares well with the observed distribution.
KEYWORDS Discrete event simulation, particle tracking, ecological modelling, modelling and simulation environment, models library, seaweeds

1 Introduction

Drifting seaweeds plays a major role in areas where they are present. They act as nursery for a vast amount of species, but also are a good bio-indicator of water quality. Such algae hosts moving ecosystems, and identifying the drift path is necessary in order to clarifying their roles. Particle tracking techniques have proven to be the most robust and common approach to identify drifting paths [Reynolds(2003)]. Mathematical models of particle tracking in the sea faces two main difficulties, the first is the lack of field data in order to evaluate the accuracy of such models. The second is the availability of ocean current data representing seasonal variability. These facts suggest that models should only retain such algorithm able to represent the minimum physics needed to reproduce the phenomena and avoid cumbersome procedures that have feeble scientific support, or that involves an excessive number of adjusting parameter. However, this minimum set of algorithms should be thoroughly tested and validated. Trajectory and spreading models usually uses resolutions of Lagrange particle tracking technique [García-Martínez and Flores-Tovar(1999)], where particles simulate material transported by the water current, with various spreading mechanisms. The most common application of particle tracking spread models is the simulation of oil spills. Various works [García-Martínez and Flores-Tovar(1999)], [Bennett and Clites(1987)] shows that the common Euler approach used to solve Lagrange equations is very sensible to time stepping and might induce numerical errors that causes erroneous drift and exaggerate the spread. This report provides a numerical experiment result that confirms the error introduced by a discrete time approach to an Euler resolution, and introduces a discrete event approach that is more suited to the simulation of ecological systems. The JeoSim model consists of two subsystems. The first is a three dimensional, sigma-coordinate hydrodynamic Model, the Princeton Ocean Model that calculates ocean velocity fields in the area of interest. The second subsystem is a Lagrangian particle tracking spreading model that calculates material trajectories using the results from the hydrological model.

Moreover, the use of a fixed time step during the simulation causes coupling problems in case of integration into a cross scale ecological model. Unlike oil spills models, drifting seaweeds has many different aspects of interest. Such models applied to ecological science requires an advanced software framework to be build upon, because the simulation is not one of an inert particle, but of a living organism with a behavior that evolves in time and influences the drifting path. Model is to be observed by different point of views, with various variables that must be taken into account (biomass, distribution, mortality,…). These require the integration of several temporal, geographical and abstraction scales, therefore using thoroughly engineered software architecture to match with the desired complexity and flexibility.

The application presented in this report is the analysis and simulation of *Sargassum* Sea grass distribution in East China Sea to clarify ecological roles of this specie. Recently a lot of drifting seaweeds were found in the East China Sea surrounded by Taiwan, China and Japan in spring. The Kuroshio Current passes north-easterly along the surface layer above 200 m isobaths

in the East China Sea. Tsushima Current is bifurcated southwest of Kyushu Island from the Kuroshio Current and enters the Japan Sea through Tsushima Straight. Origin of *Sargassum* species newly found as drifting seaweeds are thought to be geographically limited around Chinese coast in the East China Sea. *Sargassum* species were collected during on-site studies near Chinese coasts in East China Sea and Yellow Sea, and the zone further delimited with satellite imagery. Nevertheless, it is important to verify those assumptions with the use of simulation software to identify particle paths, distribution and the timing of the distribution. We will first describe the software architecture, before providing details about the underlying algorithms and finish by presenting the simulation experiment in East China Sea.

2 Material and Method

The JeoSim Software needs the pre-processing of Ocean Currents with an Ocean Circulation Model. Because of the high processing power required by such models, it is often necessary to use a supercomputer. Most ocean Circulation models are written in Fortran [Mellor and Yamada(1982)], with a few opportunities to integrate them into a monolithic software. Most of those models also offers routines for particle tracking [Pacanowski *et al.(1991)*] [Mellor and Yamada(1982)], but are not very convenient because the tracking calculations are usually made synchronously with the flow calculation. Given that an ocean flow simulation can take as much as one month on a supercomputer. Using pre-processed current flows provides a more flexible way to perform simulation of different scenarios. The separation between the tracking and the flow calculation also enables the use of any other data sets. To facilitate the use of the ocean circulation model, JeoSim provides a software interface for management of ocean model initialization data (bathymetry, temperature, salinity fields and wind stress). The biggest originality of the software lies in the particle tracking module. Because the purpose of this software is to simulate biological systems, Jeosim is not a straightforward Lagrangian particle tracking simulator, but includes interfaces with GIS, and the possibility to integrate particle tracking into cross scale models.

2.1 Modelling methodology

Jeosim is a tool designed for ecosystem modellers. Therefore, the user should focus only on modelling, while the associated simulation is generated automatically.

To do so, it is important to base the separation between modelling and simulation on a strong formal basis. This ensures that the model can be verified analytically, and that the translation between the model and the simulation is mathematically correct.

The mathematical foundations of our discrete events approach lies in Systems Theory. The basics has been developed by Zeigler in the discrete event system specification (DEVS) formalism [Zeigler(1990)]. In this formalism, models are constructed in a modular and hierarchical manner. DEVS formalism is based on the use of two kinds of models: basic and coupled. Basic models describe the behavior of basic system entities. Coupled models correspond to the structure and contain basic and coupled models as well as the link between them. Both basic and coupled models can be used as model components.

A basic DEVS model *BM* is a structure :

$BM = \langle X, S, Y, d_{\text{int}}, \mathbf{t}_a, d_{ext}, \lambda \rangle$, where:

- X: $\{(p,v) | (p \in \text{input ports}, v \in Xp)\}$ is the set of input ports and values for the reception of external events,
- Y: $\{(p,v) | (p \in \text{output ports}, v \in Yp)\}$ is the set of output ports and values for the emission of events,
- S is the set of internal sequential states,
- $d_{\text{int}} : S \rightarrow S$ is the internal transition function that will move the system to the next state after the time returned by the time advance function,
- $\mathbf{t}_a : S \rightarrow R^+$ is the time advance function that will give the life time of the current state (returns the time to the next internal transition),
- $d_{ext} : Q \times X \rightarrow S$ is the external transition function that will schedule the states changes in reaction to an input event,
- $\lambda : Q \times X \rightarrow S$ is the output function that will generate external events just before the internal transition takes place.

The dynamic interpretation is the following:

- $Q = \{(s,e) | (s \in S, 0 < e < ta(s)\}$ is the total states set and parameter.
- e is the elapsed time since last transition, and s the partial set of states for the duration of $ta(s)$ if no external event occur.
- δ_{int} : the model being in a state s at ti, it will go into s', $s' = \delta_{int}(s)$, if no external events occurs before $ti + ta(s)$.

- δ_{ext} : when an external event occurs, the model being in the state *s* since the elapsed time *e* goes in *s'*, $s' = \delta_{ext}(s,e,x)$.
  - The next state depends on the elapsed time in the present state.
  - At every state change, *e* is reset to 0.
- λ : the output function is executed before an internal transition, before emitting an output event the model remains in a transient state. A state with an infinite life time is a passive state (steady state), else, it is an active state (transient state). If the state *s* is passive, the model can evolve only with an input event occurrence.

The coupled DEVS model CM is a structure :
$CM = < X, Y, D, \{ M_d \in D \}, EIC, EOC, IC >$, where:

- *X* is the set of input ports for the reception of external events,
- *Y* is the set of output ports for the emission of external events,
- *D* is the set of components (coupled or basic models),
- M_d is the DEVS model for each $_d \in D$,
- *EIC* is the set of input links, that connects the inputs of the coupled model to one or more of the inputs of the components that it contains,
- *EOC* is the set of output links, that connects the outputs of one or more of the contained components to the output of the coupled model,
- *IC* is the set of internal links, that connects the output ports of the components to the input ports of the components in the coupled models.

In a coupled model, an output port from a model $M_d \in D$ can be connected to the input of another $M_d \in D$ but cannot be connected directly to itself. Both coupled and basic models can stand alone and are stored in a models library for reuse and archiving.

A very important point for ecosystem modellers is that models specified in DEVS can be automatically simulated. To each component of a given model is attached a generic simulation component. The simulation process is driven by messages exchange between ports in order to generate output events from inputs of the studied system. In itself, the formalism is not adapted to suit specific needs, but it can easily be extended. Next section presents the adaptation of DEVS for spatially explicit models such as treated by JeoSim.

2.2 DEVS formalism adapted to spatially explicit models

The adapted method to suit spatially explicit models method consists in modelling each component of the studied system separately [Acquaviva *et al.(2004)*]. The system is represented with thematic layers communicating if necessary for the simulation. This is similar to the GIS representation of a system where several layers, each representing a specific point of view of the reality, are superposed to represent the whole system. With this representation, a specialist of each discipline involved in the study can work in his own field of knowledge and model a part of the system. In this approach, the models that are developed are just automatically coupled. It implies an efficient tasks distribution, a simplified reuse of existing models and a more targeted reengineering process, see [Filippi and Bisambiglia(2003)] for details about the software. Figure 1 represents the Object Oriented Architecture of the Software. In the specific problem a particle derives from a GeoAtomic model, which is a model with spatial attributes. Each particle/geoatomic model is associated to a simulator and the coupled models that contain those Geoatomics also have data layers that drive the simulation.

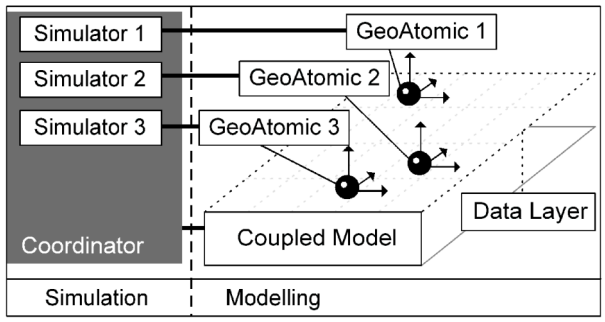


Figure 1: Architecture of the system

Thanks to the use of a strong theoretical base and the open architecture, JeoSim can be easily extended to include other biological processes that could occur at different time and space scales. The next section presents in detail such adaptation for the case of the algae model.

2.3 The DEVS *Sargassum* algae model

A patch of algae is treated as a particle, that has a specific location, area and biomass. It has the following DEVS structure :

- X: {(*p*,*v*)|(p{*speed*}, v{*x-speed*,*y-speed*,*z-speed*})}
- Y: {(*p*,*v*)|(p{position, status},*v*{x, y, z}{attached, floating, sinking})}
- *S*:{*attached*,*floating*,*sinking*},*parameters*{*x*,*y*,*z*,*biomass*,*area-size*} ,
- δ_{int} : $S \rightarrow S$ updates position, and move to the next sequential state if appropriate.
- $\mathbf{t}_a$: $S \rightarrow R^+$ returns the time to next position update, or next state change,
- δ_{ext} : $Q \times X \rightarrow S$ receives new speed information in case of position or state update,
- λ : $Q \times X \rightarrow S$ outputs new position and state. Next section presents the particle tracking algorithm adapted for the algae dispersion model.

Next section presents the particle tracking algorithm adapted for the algae dispersion model.

2.4 Lagrangian discrete event particle-tracking trajectory model

JeoSim calculates trajectories based on the assumption that behavior of drifting Seaweeds, like oils pills, can be idealized by a large number of particles that moves in three dimensions in a water body by advection and spreading dynamics.

To determine the advective velocity field, JeoSim uses the 3D sigma coordinates Princeton Ocean Model [Mellor and Yamada(1982)], that simulates ocean current, surface elevation, temperature, salinity and wind stress. The Princeton Ocean Model has been extensively tested and validated for the simulation of sea currents problems [POM(2005)].

The velocities resulting from the hydrodynamic model is later used by the spread model to compute trajectories. The Lagrange particle tracking trajectory algorithm is based on the following vector equation:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_a(\mathbf{x}_i, t) + \mathbf{v}_d(\mathbf{x}_i, t)$$

Where $\mathbf{x}_i$ is the I particle coordinate, $\mathbf{v}_a$ the advective velocity at the particle coordinate, and $\mathbf{v}_d$ is the random velocity fluctuation. The hydrodynamic model determines the advective velocities at discrete times, tn for t = 0..n and store it in a NetCDF [Unidata(2004)] file for easy storing and analysis. The method most frequently used to solve is the system of Ordinaries differential equations is the Euler method. Generally, it is implemented in the following way:

$$\mathbf{x}_i^{n+1} \cong \mathbf{x}_i^n + \Delta t[\mathbf{v}_a(\mathbf{x}_i^n, t^n) + \mathbf{v}_d(\mathbf{x}_i^n, t^n)]$$

Where Δt is the time step, and n is the time index such that $tn = n\Delta t$. This method is first order accurate, and particle tracking trajectories may diverge greatly from real ones as times advances unless Dt is very small.

A very common method for achieve a better accuracy at an equivalent time stepping, is to use a trapezoidal approximation, which, uses a prediction-correction scheme such as:

(predictor) $\mathbf{x}_i^{n+1*} = \mathbf{x}_i^n + \Delta t[\mathbf{v}_a(\mathbf{x}_i^n, t^n)]$

(corrector) $\mathbf{x}_i^{n+1} \cong \mathbf{x}_i^n + \frac{\Delta t}{2}[\mathbf{v}_a(\mathbf{x}_i^n, t^n) + \mathbf{v}_a(\mathbf{x}_i^{n+1*}, t^{n+1}) + \mathbf{v}_d(\mathbf{x}_i^n, t^n) + \mathbf{v}_d(\mathbf{x}_i^{n+1*}, t^{n+1})]$

Nevertheless, in highly non uniform flows, such as found in oceans, this method stills require a relative small time step to keep a good accuracy in zones of high speed currents. One way to overcome this limitation is the use of discrete event simulation, where there is no fixed time step, and the calculation is made instead for a step size Δq. Here n is an event index, and corresponds to an iteration with n = 0,1,2,3,.. being trajectories iterations indices. At each iteration, the time advance estimate ta is computed to be the time taken by the particle to travel the distance Delta q. After each iterations, local time at the particle is updated to be tnext = t+ta.

With the trapezoidal rules, this gives the following implementation:

(step predictor)

$$t_a^* = \frac{\Delta q}{\mathbf{v}_a(\mathbf{x}_i^n, t)}$$

(predictor)

$$\mathbf{x}_i^{n+1*} = \mathbf{x}_i^n + t_a^*[\mathbf{v}_a(\mathbf{x}_i^n, t)]$$

(step estimate)

$$t_a = \frac{2\Delta q}{\mathbf{v}_a(\mathbf{x}_i^n, t^n) + \mathbf{v}_a(\mathbf{x}_i^{n+1*}, t + t_a)}$$

(corrector)

$$\mathbf{x}_i^{n+1} \cong \mathbf{x}_i^n + \frac{t_a}{2}[\mathbf{v}_a(\mathbf{x}_i^n, t^n) + \mathbf{v}_a(\mathbf{x}_i^{n+1*}, t + t_a) + \mathbf{v}_d(\mathbf{x}_i^n, t) + \mathbf{v}_d(\mathbf{x}_i^{n+1*}, t + t_a)]$$

3 Spreading calculations

Random velocities are obtained by random sampling in the range of velocities [-Ur, Ur], [-Vr, Vr] [-Wr, Wr] to simulate spreading of seaweed patches. Those ranges are assumed to be proportional to diffusion coefficients in x, y and y. Velocities diffusion has been estimated from a study from [Bograd *et al.(1999)*], which uses satellites tracking of drifters in the northern Pacific to perform empirical estimates of diffusion coefficients. The relative diffusion coefficient is expressed by the following Formula: Du = Ur/U(rms) and Dv = Vr/V(rms). [Bograd *et al.(1999)*] found an average Du to be 0.1 and Dv = 0.08. In the model, Dw = Wr/W(rms) is given by the mortality and loss of floatability of the seaweed. Therefore, it is not assumed to have additional random motion, because the mortality distribution in itself is random.

This extensive survey also suggests that the use of that overall diffusion coefficient is of a sufficient approximation for modelling studies. The general equation for Euler with dispersion is Vd= (Vx + Vd)/ dt. To estimate Vd, we then apply the diffusion coefficient to the RMS speed obtained from the flow calculation.

3.1 Numerical Experiments

The purpose of this section is to show and calculate the difference between Euler method, trapezoidal approximation, discrete Event and discrete time. To be able to verify the results against analytical ones, we neglect the diffusion coefficients in (1) the dispersion would only be dependant of the precision algorithm and truncation errors. The numerical experiment consists of a Lagrangian resolution of the circle equation. The analytical solution in this velocity field is :

$x_e(t)=\exp(at)[x_0\cos bt - y_0 \sin bt]$

$y_e(t)=\exp(at)[x_0\sin bt - y_0 \cos bt]$

Where x0 and y0 are initial position, and xe(t) and ye(t) are position at time t.

Figure 2a) presents the velocity field for a = 0 and b = 0.001, corresponding to a circular rotating flow, and the analytical solution of the particle path at a radius of 1000.

Figure 2b) presents the tracking in discrete time with Euler using dT of 60 seconds. It is clear that the Euler method causes an outward drift of particles with an error that is dependant of the flow speed.

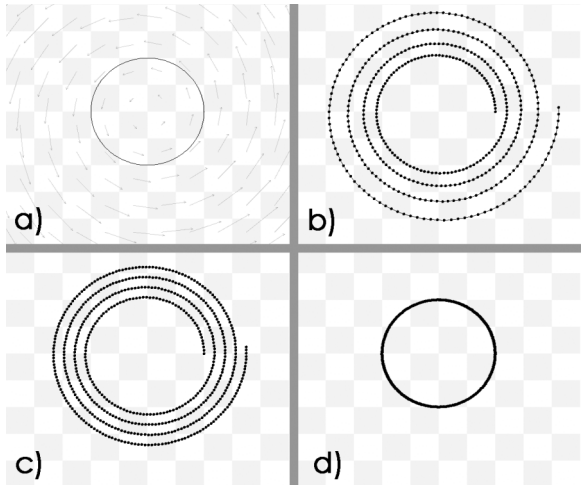


Figure 2: Particle trajectory simulation, a) Analytical solution, b) Euler resolution in discrete time, Dt = 60 seconds, c) Euler resolution in discrete event, Q = 60m, d) Euler resolution with trapezoidal interpolation, in discrete event, q = 60 meters

Figure 2c) presents the particle path with Euler, but in a discrete event Fashion, with a quantum of 60 meters, that means an original time step of 60 s (with speed = 1m/s), equal to the one shown in figure 2b). We can see from this figure, that here the error is independent of the flow speed, because whatever the flow speed is, the particle moves the same distance. This also means that the error would be high if compared with discrete time in areas where the flow speed is slow, because the calculations would be less frequent. Nevertheless using Discrete event simulation is interesting in turbulent flows, because the variability of the flow speed is high in space, but small in time. Using this technique will always keep the numerical errors within the same interval whatever the flow speed is. Numerically exaggerated diffusion is here of 500 meters after 3.5 hours, and 1200 Meters after 7 hours.

Figure 2d) presents the particle Path with the trapezoidal interpolation, in discrete event, with a quantum of 60 meters, that shows a path resolution almost identical to the analytical solution. This kind of resolution greatly limits the need for a big quantum, and is particularly well suited to be used in conjunction with discrete event simulation because the trapezoidal evaluation will be make with trapezoids of the same size, ensuring here too that the error is kept independent of the flow speed.

Euler method, like the trapezoidal approximation method provides no error in laminar flow, nevertheless in turbulent flows, such as found in Ocean or atmosphere. It is important to minimize the error and assume it the most constant over the surface to be able to quantify it, whatever the flow speed is. For it is assumed that the association of a discrete event simulation with trapezoidal Euler approximation is good tradeoffs between computational efficiency and robustness. The next section presents the application of the algorithm into marine science through the simulation of seaweed propagation in East China Sea.

4 Application and discussion

In this section, Jeosim is applied to predict *Sargasso* distribution in East China sea. The aim of the simulation experiment is to have a better understanding of the *Sargasso* Path and validate the model and diffusion coefficients thanks to observation made onboard of the RV Hako-Maru (main research boat of the Ocean Research Institute) in march 2000. The Ocean Circulation simulation has been performed using the Princeton Ocean Model, a curvilinear sigma coordinates model that is very broadly used for flow calculations. The grid is 150 cells wide, 80 high and has 20 levels. With a spacing of 34.4 minutes per cell horizontally and 27.75 vertically on the grid. The studied region is bounded by 114 degrees east, 20 degrees west for longitudes and 5 degrees north, 42 degrees north for latitude. The model has been spinned up using boundaries condition provided by NCEP current reanalysis project [NCEP(2004)]. NCEP data has been used to initialize the fields at a lower resolution, then run the simulation with flows updated at every time step from the NCEP dataset. Figure 3 presents the boundaries conditions used, the shaded area is the current data available in the NCEP dataset, while Yellow and Japan sea currents are simulated. The bottom topography is the smoothed data from the National Geophysical Data Center database with 2 minute resolution (ETOPO2). The horizontal diffusivities are modelled using the Smagorinsky (1963)[Smagorinsky(1963)] form with the 0.2 coefficient chosen for this application. The bottom stress is assumed to follow a quadratic law

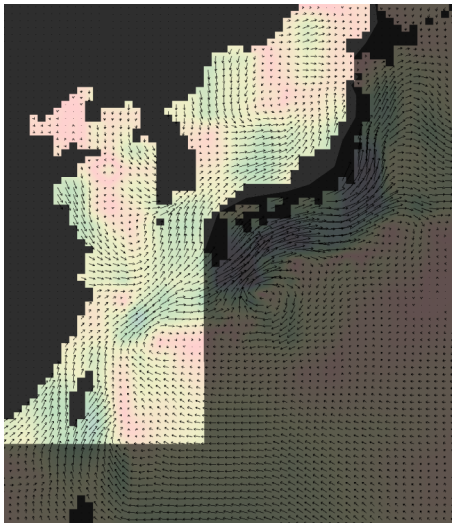
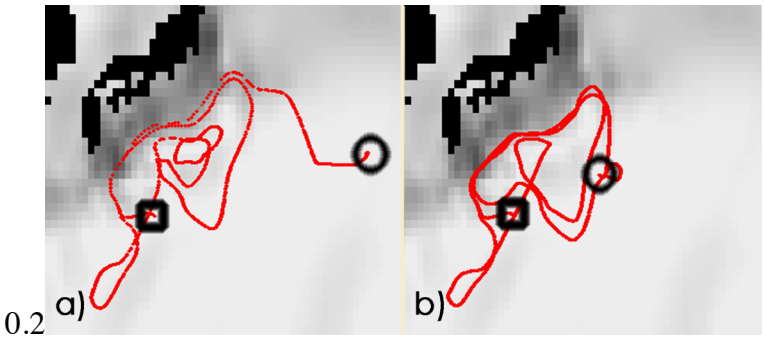


Figure 3: Simulated current (January 2001) and boundaries conditions of the Ocean current model

The currents in the shaded area are loaded and interpolated from the dataset at every time step for a period of 5 years, then the simulation is run using wind stress from NCAR reanalysis project [NCEP/NCAR(2004)], and Levitus [Levitus(1982)] Temperature and salinity field for two other years. The results retained are the calculations of the last simulated year, with weekly variability.

The particle tracking routines module is used thereafter to calculate the seaweed trajectories in the Kuroshio/Oyashio Region . Figure 4 shows an example of trajectories simulated using the two different schemes, but an equal amount of transitions. It appears that the discrete time method (Figure 4a)) is less accurate in high speed flows, resulting in an exaggerated drift. The discrete event method accuracy is independent of the current speed, and does not show an exaggerated drift (Figure 4b)). In fact, most of the transitions of the discrete time simulation occurred in very slow current areas. This region is knows to be a trap in the pacific ocean [Hurlburt *et al.(1996)*], where floating elements can stay up to two years. Using a discrete event method here might require a very small time step not to suffer from an exaggerated drift that might push the elements out of the current earlier than expected (Figure 4a)). Once diffusion is added to the model, several thousands of trajectories has to be calculated in parallel in order to represent the possible paths of the particles, for this reason, the discrete event method seems to be the most appropriate.



0.2

Figure 4: Six month trajectory simulation without diffusion, a) Euler method/discrete time, b) Trapezoidal interpolation method/ discrete event. Square represents the start position, circle the end position, shading shows the absolute flow speed.

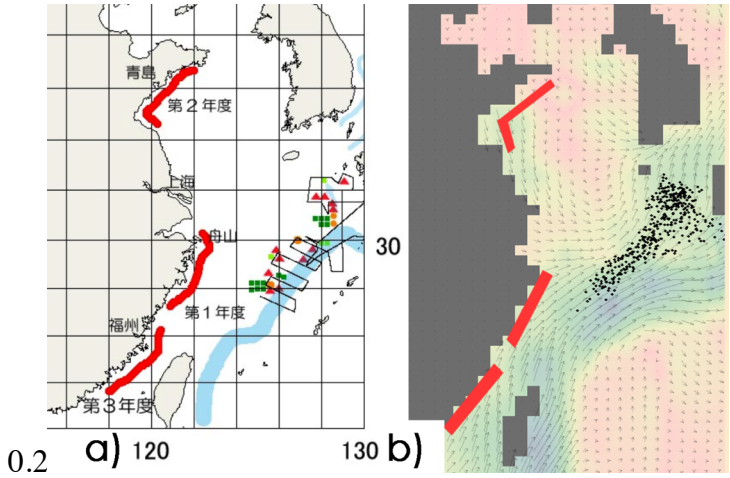


Figure 5: a) Distribution of drifting seaweeds in East China sea (circle squares and triangles maps to different *Sargassum* species). Coastal strokes correspond to the supposed *Sargassum* origin, while the marine stroke represents the hot Kuroshio Current flowing to the north east. b) Particles dispersion in East China Sea (black dots), after 75 days of simulation using Jeosim starting from March 15. Stokes corresponds to the release zones, shading represents the absolute flow speed.

The Laboratory of Behavior, Ecology and Observation Systems at the Ocean Research Institute (University of Tokyo) filed observation in the East China Sea using R/V Hakuho-Maru. They mapped geographical distribution of drifting seaweeds and collected samples of drifting seaweeds to detect the species. Figure 5a) shows the distribution of *Sargassum* in the East China Sea (triangles, squares and circles) as found by the Laboratory in May 2002. Origin of *Sargassum* species newly found as drifting seaweeds are thought to be geographically limited around Chinese coast in the East China Sea.

Figure 5b) presents the results of the simulation with a release date of 15 February, and a particle lifetime of 90 days. The numerical simulation using Jeosim provides results that prove to be very similar to the observation. In particular, it shows that the seaweeds are distributed on a path that is more north than the observation, which provides a good hint on where should be performed next observation.

7 Conclusion

We have introduced a new combination of techniques that proves to be very well suited for the modelling of drifting ecosystems. The first advantage is that it directly uses the current fields calculated from the ocean circulation model. Currents are simulated only once, and particle tracking is made using the generated vector fields. Using subroutines integrated into the Ocean Model greatly limits the number of experiments and the ease of use because each time particle tracking is performed, currents have to be re-calculated, an ocean current simulation can takes up to 2 months. The second advantage is that the software is discrete event based, and not discrete time base, that means that the simulation has no time steps. Instead of calculating the distance travelled by a given particle for a fixed time, we calculate how long it takes for a particle to travel a fixed distance (like finite elements method). From a marine ecology point of view, the outcome of the research can be rather interesting because it will allow studying the *Sargassum* species that can be rather invasive and become an ecological problem. Future versions of the software will integrate the possibility to add a more complex behavior for the particles to represent the nesting of fish under those algae.

Acknowledgements

This research is supported by the Japanese Society for the promotion of Science.

References

[Acquaviva *et al*.(2004)] Acquaviva, J., Filippi, J., and Bisgambiglia, P. (2004). A multi-layered modeling and simulation architecture. In *International Mediterranean Modeling Multiconference*". CD-ROM, GISIG, Genoa, IT.

[Bennett and Clites(1987)] Bennett, J. and Clites, A. (1987). Accuracy of trajectory calculation in a a finite-diference circulation model. *J. Comp. Phys*., **68**(2), 272–282.

[Bograd *et al*.(1999)] Bograd, S., Thomson, R., Rabinovich, A., and LeBlond, P. (1999). Near-surface circulation of the northeast pacific ocean derived from woce-svp satellite-tracked drifters. *Deep-Sea Research Part II*, **46**(11-12), 2371–2403.

[Filippi and Bisambiglia(2003)] Filippi, J.-B. and Bisambiglia, P. (2003). Jdevs :an implementation of a devs based formal framework for environmental modelling. *Environmental modelling and Software*. A paraître, disponible sur http://www.sciencedirect.com.

[García-Martínez and Flores-Tovar(1999)] García-Martínez, R. and Flores-Tovar, H. (1999). Computer modeling of oil spill trajectories with a high accuracy method. *Spill Science and Technology Bulletin*, **5**(5-6), 323–330.

[Hurlburt *et al*.(1996)] Hurlburt, H., Wallcraft, A., Schmitz, W., Hogan, P., and Metzger, E. (1996). Dynamics of the kuroshio/oyashio current system using eddy-resolving models of the north pacific ocean. *Journal of Geophysical Research*, **101**(10), 941–976.

[Levitus(1982)] Levitus, S. (1982). *Climatological atlas of the world ocean*. U. S. Government Printing Office.

[Mellor and Yamada(1982)] Mellor, G. and Yamada, T. (1982). Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys*, **20**(4), 851–875.

[NCEP(2004)] NCEP (2004). Ncep pacific ocean analysis. http://www.cdc.noaa.gov/cdc/data.ncep.pac.ocean.html.

[NCEP/NCAR(2004)] NCEP/NCAR (2004). The ncep/ncar reanalysis project. http://www.cdc.noaa.gov/cdc/reanalysis/.

[Pacanowski *et al*.(1991)] Pacanowski, R. C., Dixon, K., and Rosati, A. (1991). The gfdl modular ocean model user guide. *Technical Report, Princeton Geophysical Fluid Dynamics Laboratory*, **1**(2), 16.

[POM(2005)] POM (2005). Priceton ocean model homepage, and pom applications. http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/.

[Reynolds(2003)] Reynolds, A. (2003). On the nonextensive statistical mechanics of tracer-particle motions in turbulence. *Physical Review Letters*, **91**. article 84503.

[Smagorinsky(1963)] Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, **91**, 99–164.

[Unidata(2004)] Unidata (2004). Network common data form. http://my.unidata.ucar.edu/content/software/netcdf/.

[Zeigler(1990)] Zeigler, B. (1990). *Object-Oriented Simulation with Hierarchical, Modular Models*. Academic Press.