A Particle-Mesh Numerical Method for A.D.R. Equations with Applications to Plankton Modelling

F. Oliveri & F. Paparella
Dip. di Matematica, Università di Messina
Dip. di Matematica “E. De Giorgi”, Università di Lecce

WASCOM 2007 - Scicli
Plankton Seen from the Sky

From: http://disc.sci.gsfc.nasa.gov/oceancolor/scifocus/classic_scenes/00_classics_index.shtml
Main Factors Affecting Plankton Dynamics

- **Population Dynamics**: predators (zooplankton), preys (phytoplankton), nutrients (nitrates, phosphates, etc.).

- **Stirring by Ocean Currents**: generally chaotic, possibly inhomogeneous (vortices), essentially two-dimensional.

- **Small-Scale Mixing**: small scale turbulence, wave-induced drift, swimming!

- **Space-Time Inhomogeneities of Parameters**: upwelling (sudden burst of nutrients), weather (changes temperature, salinity, sunlight).
Main Factors Affecting Plankton Dynamics

- **Population Dynamics**: predators (zooplankton), preys (phytoplankton), nutrients (nitrates, phosphates, etc.).

- **Stirring by Ocean Currents**: generally chaotic, possibly inhomogeneous (vortices), essentially two-dimensional.

- **Small-Scale Mixing**: small scale turbulence, wave-induced drift, swimming!

- **Space-Time Inhomogeneities of Parameters**: upwelling (sudden burst of nutrients), weather (changes temperature, salinity, sunlight).
Main Factors Affecting Plankton Dynamics

- **Population Dynamics**: predators (zooplankton), preys (phytoplankton), nutrients (nitrates, phosphates, etc.).

- **Stirring by Ocean Currents**: generally chaotic, possibly inhomogeneous (vortices), essentially two-dimensional.

- **Small-Scale Mixing**: small scale turbulence, wave-induced drift, swimming!

- **Space-Time Inhomogeneities of Parameters**: upwelling (sudden burst of nutrients), weather (changes temperature, salinity, sunlight).
Main Factors Affecting Plankton Dynamics

- **Population Dynamics**: predators (zooplankton), preys (phytoplankton), nutrients (nitrates, phosphates, etc.).

- **Stirring by Ocean Currents**: generally chaotic, possibly inhomogeneous (vortices), essentially two-dimensional.

- **Small-Scale Mixing**: small scale turbulence, wave-induced drift, swimming!

- **Space-Time Inhomogeneities of Parameters**: upwelling (sudden burst of nutrients), weather (changes temperature, salinity, sunlight).
The concentration $c_i$ of the $i-$th species is described by

$$\frac{\partial c_i}{\partial t} + J(\psi, c_i) = D_i \Delta c_i + f_i(c_1, \ldots, c_N)$$

where

- $\psi$ is the streamfunction (the velocity field is $u = (-\partial_y \psi, \partial_x \psi)$)
- $J$ is the Jacobian determinant ($J(A, B) = \partial_x A \partial_y B - \partial_y A \partial_x B$)
- $D_i$ is the diffusion coefficient of the $i-$th species.
- $f_i$ is the $i-$th component of the vector field determining the population dynamics.
Advection-Reaction Equations Are Numerically Easy

Just solve a system of ODEs along each characteristic line:

$$\frac{Dc_i}{Dt} = f_i(c_1, \ldots, c_N)$$

The flow is *volume-preserving*: if the initial conditions sample uniformly the domain, the sample will remain uniform at any later time.
Particle Algorithm for A-R Equations

- Seed with uniform random probability $M$ particles in the domain.
- Assign to the $p$–th particle the $N$ concentrations $c_{p,1}(0), \ldots, c_{p,N}(0)$.
- Numerically solve the following ODEs

\[
\begin{align*}
\dot{x}_p &= -\frac{\partial \psi}{\partial y} \\
\dot{y}_p &= \frac{\partial \psi}{\partial x} \\
& \vdots \\
\dot{c}_{p,i} &= f_i(c_{p,1}, \ldots, c_{p,N}) \\
& \vdots
\end{align*}
\]
Particle Algorithm for A-R Equations

- Seed with uniform random probability $M$ particles in the domain.
- Assign to the $p$–th particle the $N$ concentrations $c_{p,1}(0), \ldots, c_{p,N}(0)$.
- Numerically solve the following ODEs

$$
\begin{align*}
\dot{x}_p &= -\frac{\partial \psi}{\partial y} \\
\dot{y}_p &= \frac{\partial \psi}{\partial x} \\
&\vdots \\
\dot{c}_{p,i} &= f_i(c_{p,1}, \ldots, c_{p,N}) \\
&\vdots
\end{align*}
$$
Particle Algorithm for A-R Equations

- Seed with uniform random probability $M$ particles in the domain.
- Assign to the $p$–th particle the $N$ concentrations $c_{p,1}(0), \ldots, c_{p,N}(0)$.
- Numerically solve the following ODEs

\[
\begin{align*}
\dot{x}_p &= -\frac{\partial \psi}{\partial y} \\
\dot{y}_p &= \frac{\partial \psi}{\partial x} \\
&\vdots \\
\dot{c}_{p,i} &= f_i(c_{p,1}, \ldots, c_{p,N}) \\
&\vdots
\end{align*}
\]
The 'cloud' \( P \) is a function with compact support, normalized, symmetric and unimodal. The particle in \( x_p \) affects the grid node in \( x_i \) with the statistical weight

\[
w_{i,p} = \int_{x_i-\Delta/2}^{x_i+\Delta/2} P(x-x_p) \, dx
\]

The concentration \( c_p \) carried by the particles is averaged on the mesh nodes as

\[
\tilde{c}(x_i) = \frac{\sum_p w_{i,p} c_p}{\sum_p w_{i,p}}
\]
Diffuse on the Mesh, then Relax to the Particles

Jacobi sweep:

\[
\tilde{c}(x_{i,j}) = \frac{\tilde{c}(x_{i+1,j}) + \tilde{c}(x_{i-1,j}) + \tilde{c}(x_{i,j+1}) + \tilde{c}(x_{i,j-1})}{4}
\]
Diffuse on the Mesh, then Relax to the Particles

Jacobi sweep:

\[
\tilde{c}(x_{i,j}) = \frac{\tilde{c}(x_{i+1,j}) + \tilde{c}(x_{i-1,j}) + \tilde{c}(x_{i,j+1}) + \tilde{c}(x_{i,j-1})}{4}
\]

The concentration field of each particle relaxes towards the gridded field with a time constant inversely proportional to the diffusivity \(D\):

\[
\dot{c}_p = \frac{4D}{\Delta^2} (\tilde{c}(x_p) - c_p)
\]

where \(\tilde{c}(x_p)\) is the gridded field interpolated at the position of the \(p\)-th particle and \(\Delta\) is the mesh size.
Diffuse on the Mesh, then Relax to the Particles

Jacobi sweep:

\[
\tilde{c}(x_{i,j}) = \frac{\tilde{c}(x_{i+1,j}) + \tilde{c}(x_{i-1,j}) + \tilde{c}(x_{i,j+1}) + \tilde{c}(x_{i,j-1})}{4}
\]

The concentration field of each particle relaxes towards the gridded field with a time constant inversely proportional to the diffusivity \(D\):

\[
\dot{c}_p = \frac{4D}{\Delta^2} (\tilde{c}(x_p) - c_p)
\]

where \(\tilde{c}(x_p)\) is the gridded field interpolated at the position of the \(p\)–th particle and \(\Delta\) is the mesh size.

If \(D \to 0\) we recover the particle algorithm for A-R equations!
The “Renovating Random Wave Model” uses the following streamfunction

$$\psi(x, y, t) = \frac{U}{k} [p_i \sin(kx + \phi_i) - q_i \sin(ky + \phi_i)]$$

where $i = \text{int}(2 \times t/\tau)$; $\phi_i$ is uniformly random in $[0, 2\pi)$; $p_i$ is one for even $i$ and zero otherwise; $q = 1 - p$. 
Predator-Prey Model (from May 1972)

\( P \) is concentration of **prey**
\( Z \) is concentration of **predator**

\[
\begin{align*}
\frac{dP}{dt} &= P(1 - P) - aZ(1 - e^{-\lambda_1 P}) \\
\frac{dZ}{dt} &= -\gamma Z + bZ(1 - e^{-\lambda_2 P})
\end{align*}
\]

Stable limit cycle for a wide range of parameters. Here we use
\( a = 1; \ b = 2.5; \ \gamma = 1.5; \ \lambda_1 = \lambda_2 = 4.0 \)
"Checkerboard" Streamfunction on a $L_x \times L_y$ domain:

$$\psi = A \sin(kx) \sin(ky)$$

with $A = 0.1$, $k = 1.5$, $L_x = L_y = 2\pi$. 
"Checkerboard" Streamfunction on a $L_x \times L_y$ domain:

$$\psi = A \sin(kx) \sin(ky)$$

with $A = 0.1$, $k = 1.5$, $L_x = L_y = 2\pi$.

Initially the domain is full of preys, with a few predators in a small patch inside a vortex.
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

\[ D = 0 \]

\[ D = 10^{-7} \]
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

\[ D = 0 \]

\[ D = 10^{-7} \]
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$$D = 0$$

$$D = 10^{-7}$$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

\[ D = 0 \]

\[ D = 10^{-7} \]
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Is Diffusion Really Important?

$D = 0$

$D = 10^{-7}$
Ever Since Huygens...

...it has been known that coupled oscillators may synchronize.
Ever Since Huygens...

...it has been known that coupled oscillators may **synchronize**.

Plankton in the oceans may be seen as a collection of (infinitely many) oscillators with near-neighbour diffusive coupling.
Ever Since Huygens...

...it has been known that coupled oscillators may *synchronize*.

Plankton in the oceans may be seen as a collection of (infinitely many) oscillators with near-neighbour diffusive coupling.

Stirring changes the identity of the neighbours....
Ever Since Huygens...

...it has been known that coupled oscillators may synchronize.

Plankton in the oceans may be seen as a collection of (infinitely many) oscillators with near-neighbour diffusive coupling.

Stirring changes the identity of the neighbours....

Does synchronism still emerge?
Synchronization Times

The graph illustrates the relationship between synchronization time and diffusivity. The x-axis represents diffusivity ranging from $10^{-8}$ to $10^{-5}$, while the y-axis represents synchronization time ranging from $10^2$ to $10^4$. The shaded area indicates the region of incomplete synchronization.
We have proposed a particle-mesh algorithm for A-D-R equations which recovers the non diffusive limit.
Conclusions

- We have proposed a particle-mesh algorithm for A-D-R equations which recovers the non diffusive limit.

- For small diffusivities plankton synchronization is attained at extremely long time scales (if ever). In the oceans, at those scales other phenomena come into the picture.
Conclusions

- We have proposed a particle-mesh algorithm for A-D-R equations which recovers the non diffusive limit.

- For small diffusivities plankton synchronization is attained at extremely long time scales (if ever). In the oceans, at those scales other phenomena come into the picture.

- What happens for chaotic (rather than periodic) population dynamics?
Conclusions

- We have proposed a particle-mesh algorithm for A-D-R equations which recovers the non diffusive limit.

- For small diffusivities plankton synchronization is attained at extremely long time scales (if ever). In the oceans, at those scales other phenomena come into the picture.

- What happens for chaotic (rather than periodic) population dynamics?

- What happens for non-homogeneous flows (e.g. with vortices, etc.)?
Conclusions

- We have proposed a particle-mesh algorithm for A-D-R equations which recovers the non diffusive limit.

- For small diffusivities plankton synchronization is attained at extremely long time scales (if ever). In the oceans, at those scales other phenomena come into the picture.

- What happens for chaotic (rather than periodic) population dynamics?

- What happens for non-homogeneous flows (e.g. with vortices, etc.)?