A parallel algorithm for solving a multi-layer Convection-Diffusion Problem

L. Ferragut, M.I. ASENSIO
Instituto de Física Fundamental y Matemáticas y Departamento de Matemática Aplicada, Universidad de Salamanca
ferragut@usal.es, mas@usal.es

J. M. Cascón
Instituto de Física Fundamental y Matemáticas y Departamento de Economía e Historia Económica, Universidad de Salamanca
casbar@usal.es

D. Prieto
Departamento de Matemática Aplicada, Universidad de Salamanca
dpriher@usal.es

Abstract

We apply the general theory of splitting algorithms to the construction of a multi-layer convection-diffusion model and its numerical approximation based on a combination of the Adaptive Finite Element Method with characteristics in the horizontal directions and Finite Differences in the vertical direction. We present a parallel version of the algorithm. We show the parallel efficiency of the algorithms through several examples.

Sección en el CEDYA 2011: AN

1. General splitting algorithms of evolution equations

The splitting methods are widely used in various kinds of problems. Furthermore, parallel computing provides the tools to realize the idea behind this kind of methods: splitting complicated problems into simpler systems. For a complete review of these methods see [1]. In this work we focused in operator splitting methods and product formula algorithms for general equations of evolution. For a brief introduction of this topic, see [2].

Let consider the following general equation of evolution, which can be used to mathematically represent a wide variety of environmental problems,

\[ M \frac{du}{dt} + A(u) = f, \]  \hspace{1cm} (1)

\[ u(0) = u_0, \] \hspace{1cm} (2)
where $u$ is a $d$-dimensional vector, $M$ is a positive definite symmetric matrix and $A$ is a function (not necessary linear) from $\mathbb{R}^d$ into $\mathbb{R}^d$.

In the following, we endow $\mathbb{R}^d$ with the following energy inner product 

\[
(Mu,v) \quad \text{for all } u,v \in \mathbb{R}^d, \text{ with its associated norm } ||u|| = (Mu,u)^{1/2}.
\]

In this context, an unconditionally stable algorithm for equation (1)-(2) is given by a one-parameter family of functions $F(\Delta t) : \mathbb{R}^d \rightarrow \mathbb{R}^d, \Delta t > 0$, satisfying,

1. consistency:

\[
\lim_{\Delta t \to 0} M \frac{F(\Delta t)u - u}{\Delta t} = -Au + f \quad \forall u \in \mathbb{R}^d
\]

In the following methods, we will consider at least order one, that is, verifying

\[
F(\Delta t)u = u + \Delta tM^{-1}(-A(u) + f) + O(\Delta t^2)
\]

2. unconditional stability:

\[
||F(\Delta t)u - F(\Delta t)v|| \leq ||u - v|| \quad \forall u,v \in \mathbb{R}^d, \Delta t > 0
\]

When the mapping $F()$ is linear, the stability condition (5) reduces to

\[
||F(\Delta t)u|| \leq ||u|| \quad \forall u \in \mathbb{R}^d
\]

Examples

1. The Euler implicit algorithm is:

\[
M \frac{u^{n+1} - u^n}{\Delta t} + Au^{n+1} = f^{n+1}
\]

\[
u^0 = u_0
\]

in this case

\[
F(\Delta t)u = (M + \Delta tA)^{-1}u
\]

and the solution at time step $n + 1$ is given as a function of $u^n$ by

\[
u^{n+1} = (M + \Delta tA)^{-1}u^n + \Delta t(M + \Delta tA)^{-1} f^{n+1}
\]

The Euler method is consistent of order one as it is easy to verify.

2. The Crank-Nicolson scheme is:

\[
M \frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}(Au^{n+1} + Au^n) = \frac{1}{2}(f^{n+1} + f^n)
\]

\[
u^0 = u_0
\]

so we have

\[
F(\Delta t)u = (M + \frac{1}{2}\Delta tA)^{-1}(M - \frac{1}{2}\Delta tA)u
\]

which is a second order algorithm.
In a variety of environmental problems the operator \( A \) and the source term \( f \) admit an additive decomposition

\[
A = \sum_{i=1}^{N} A_i, \quad f = \sum_{i=1}^{N} f_i \quad (14)
\]

We are concerned with algorithms that exploit the additive form of \( A \) and \( f \). Let \( F_{\Delta t}^i, i = 1, \ldots, N \) denote stable algorithms consistent with \( M \) and \( A_i \). The corresponding splitting algorithm then takes the form

\[
F(\Delta t) = F_N(\Delta t)F^{N-1}(\Delta t) \ldots F_1(\Delta t) = \prod_{i=1}^{N} F_i(\Delta t) \quad (15)
\]

In other words, the algorithm \( F(\Delta t) \) amounts to applying the individual algorithms \( F_i(\Delta t) \) consecutively to the solution vector, taking the result from each one as the initial conditions for the next algorithm. The global algorithm is complete for a given time step when the individual algorithms have been applied. We have the following results about the consistency and unconditional stability of the global algorithm.

**Proposition**

The algorithm (15) is consistent with \( M \) and \( A \), and unconditionally stable, if all the individual algorithms are as well.

**Proof of consistency:** The consistency of the individuals operators \( F_i() \) implies

\[
F_i(\Delta t)u = u + \Delta t M^{-1} (-A_i(u) + f_i) + O(\Delta t^2), \quad i = 1, \ldots, N.
\]

Taken the product of \( F_1(\Delta t) \) and \( F_2(\Delta t) \) and retaining terms up to second order we obtain

\[
F_2(\Delta t)F_1(\Delta t)u = F_2(\Delta t)(F_1(\Delta t)u) = F_1(\Delta t)u - \Delta t M^{-1} A_2(F_1(\Delta t)u) + \Delta t M^{-1} f_2 + O(\Delta t^2)
\]

Proceeding by induction it is readily shown that

\[
F(\Delta t)u = (\Pi_{i=1}^{N} F_i(\Delta t))u = u - \Delta t M^{-1} \left( \sum_{i=1}^{N} A_i \right) u + \Delta t M^{-1} \left( \sum_{i=1}^{N} f_i \right) + O(\Delta t^2)
\]

\[
= u - \Delta t M^{-1} A(u) + \Delta t M^{-1} f + O(\Delta t^2)
\]

**Proof of stability:** It follows from the definition of unconditional stability (6) that for all \( u, v \in R^d \) and \( \Delta t > 0 \)

\[
||F(\Delta t)u - F(\Delta t)v|| = ||(\Pi_{i=1}^{N} F_i(\Delta t))u - (\Pi_{i=1}^{N} F_i(\Delta t))v||
\]

\[
= ||F_N(\Delta t)(\Pi_{i=1}^{N-1} F_i(\Delta t))u - F_N(\Delta t)(\Pi_{i=1}^{N-1} F_i(\Delta t))v||
\]

\[
\leq ||(\Pi_{i=1}^{N-1} F_i(\Delta t))u - (\Pi_{i=1}^{N-1} F_i(\Delta t))v||
\]
Proceeding by induction one finds

\[ \| F(\Delta t)u - F(\Delta t)v \| \leq \| u - v \| \]  \hspace{1cm} (16)

**Remark:** Although the algorithms corresponding to \( F_i(\Delta t) \) are second order accurate, the splitting algorithm (15) is not ([2]). Using a double pass procedure, the second order accuracy can be recovered, i.e.,

\[ F(\Delta t)u = (\Pi_{i=N}^1 F_i(\frac{1}{2}\Delta t))(\Pi_{i=1}^N F_i(\frac{1}{2}\Delta t))u \]  \hspace{1cm} (17)

2. A Multi-layer Convection-Diffusion Model

In the following we deal with a mathematical model of a convection-diffusion process in a three-dimensional domain corresponding to an air layer over a ground surface \( S \) not necessarily plane. Let \( \omega \subset \mathbb{R}^2 \) be a two-dimensional normalized bounded and connected domain representing the projection of the three-dimensional ground surface \( S \), let \( x = (x, y) \) be any of its points and \( t \) the time. We denote \( \Omega = \{(x, z) : x \in \omega, h(x) < z < \bar{h}\} \) to represent the air layer studied. Let \( \delta \) be the height of the domain \( \Omega \) such that the height \( h(x) \) of the surface at point \( x \) is smaller than \( \delta \). We decompose the boundary of \( \Omega \) into \( \partial \Omega = S \cup A \cup L \), where \( S = \{(x, z) : x \in \omega, z = h(x)\} \) is the ground surface, \( A = \{(x, z) : x \in \omega, z = \bar{h}\} \) is the air upper horizontal boundary and \( L = \{(x, z) : x \in \partial \omega, h(x) < z < \delta\} \) is the air lateral vertical boundary.

In this section, we denote by an index \( xy \) the two-dimensional operators and by the index \( z \) the operators concerning the vertical component, no index means three-dimensional operators. We use small letters for the two-dimensional problem, and capital letters for the three-dimensional problem. We denote the air velocity \( \mathbf{U} = (U_1, U_2, U_3) \), where we distinguish the vertical velocity from the horizontal one denoting \( W = U_3, \mathbf{V} = (U_1, U_2) \).
2.1. The Convection-Diffusion model

The convection-diffusion equation governing the dispersion of pollutants in the atmosphere is

\[
\frac{\partial u}{\partial t} + \mathbf{V} \cdot \nabla u + W \frac{\partial u}{\partial z} - \nabla_{xy} \cdot (k_{xy} \nabla_{xy} u) - \frac{\partial}{\partial z} (k_z \frac{\partial u}{\partial z}) = f
\]

where \( u \) represent the amount of pollutant at each point of the domain \( \Omega \), and at each instant \( t \), and \( k \) is a diffusion coefficient.

In order the problem to be well defined we have to add boundary and initial condition. We assume the following boundary condition on the lateral boundary \( L \) of the three-dimensional domain \( D \), in order to represent the loss of pollutant on the outdoor boundary,

\[
-k_{xy} \nabla_{xy} u \cdot \nu|_L = [V \cdot \nu]^+ u \text{ on } L
\]

where \( \nu \) is the normal unit outer vector. We assume no loss on the upper and lower boundary,

\[
-k_z \frac{\partial u}{\partial z} = 0 \text{ on } A
\]

\[
-k_{xy} \nabla_{xy} u \cdot \nu|_S - k_z \frac{\partial u}{\partial z} = 0 \text{ on } S
\]

We assume that the air is initially clean, that is \( u = 0 \) at time \( t = 0 \).

\[
u|_{t=0} = 0
\]

2.2. Change of coordinates

We make a change of coordinates in order to transform the three-dimensional domain \( \Omega \) into a cuboid. The new coordinates will be

\[
\tau = t \\
\xi = x \\
\eta = y \\
\zeta = z - h(x,y)
\]

By straightforward computations for any function \( \phi = \phi(t, x, y, z) \) we have

\[
\frac{\partial \phi}{\partial \tau} = \frac{\partial \phi}{\partial t} \\
\frac{\partial \phi}{\partial \xi} = \frac{\partial \phi}{\partial x} \frac{\partial h}{\partial \xi} + \frac{\partial h}{\partial x} \frac{\partial \phi}{\partial \xi} \\
\frac{\partial \phi}{\partial \eta} = \frac{\partial \phi}{\partial y} + \frac{\partial h}{\partial y} \frac{\partial \phi}{\partial \eta} \\
\frac{\partial \phi}{\partial \zeta} = \frac{\partial \phi}{\partial z}
\]
Then the convective term becomes,

\[ \mathbf{U} \cdot \nabla u = \mathbf{V} \cdot \nabla \xi \eta u + (W - U_1 \frac{\partial h}{\partial x} - U_2 \frac{\partial h}{\partial y}) \frac{\partial u}{\partial \zeta} \]

and the diffusive term is

\[-\nabla \cdot (k \nabla u) = - \nabla \xi \eta \cdot (k \xi \eta \nabla \xi \eta u) - \frac{\partial}{\partial \zeta} \left( \left( k + k_\xi \xi \frac{\partial h}{\partial x} \right)^2 + k_\eta \eta \left( \frac{\partial h}{\partial y} \right)^2 \right) \frac{\partial u}{\partial \zeta} + \frac{\partial}{\partial \eta} \left( k_\xi \xi \frac{\partial h}{\partial x} \frac{\partial u}{\partial \xi} \right) + \frac{\partial}{\partial \zeta} \left( k_\eta \eta \frac{\partial h}{\partial y} \frac{\partial u}{\partial \eta} \right) + \frac{\partial}{\partial \eta} \left( k_\xi \xi \frac{\partial h}{\partial \eta} \frac{\partial u}{\partial \xi} \right) + \frac{\partial}{\partial \zeta} \left( k_\eta \eta \frac{\partial h}{\partial \eta} \frac{\partial u}{\partial \eta} \right) \]

(23)

Consequently the equations in the transformed domain are analogous to (18) and (19) replacing \( W \) by \( W - V_1 \frac{\partial h}{\partial x} - V_2 \frac{\partial h}{\partial y} \) and \( k_z \) by \( k + k_\xi \xi \frac{\partial h}{\partial x} + k_\eta \eta \left( \frac{\partial h}{\partial y} \right)^2 \) plus the terms with crossed derivatives in (23). In order to simplify notation, we use again the notation \( x, y, z \) and \( t \) for the new coordinates instead of \( \xi, \eta, \zeta \) and \( \tau \), so equation (18) is now,

\[ \frac{\partial u}{\partial t} + \mathbf{V} \cdot \nabla_{xy} u + (W - V_1 \frac{\partial h}{\partial x} - V_2 \frac{\partial h}{\partial y}) \frac{\partial u}{\partial z} - \nabla_{xy} \cdot (k_{xy} \nabla_{xy} u) - \frac{\partial}{\partial z} \left( k + k_{xy} \left( \frac{\partial h}{\partial x} \right)^2 + k_\eta \eta \left( \frac{\partial h}{\partial y} \right)^2 \right) \frac{\partial u}{\partial \zeta} + \frac{\partial}{\partial x} \left( k_{xy} \frac{\partial h}{\partial x} \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial \zeta} \left( k_{xy} \frac{\partial h}{\partial x} \frac{\partial u}{\partial \xi} \right) + \frac{\partial}{\partial y} \left( k_{xy} \frac{\partial h}{\partial y} \frac{\partial u}{\partial \zeta} \right) + \frac{\partial}{\partial \zeta} \left( k_{xy} \frac{\partial h}{\partial y} \frac{\partial u}{\partial \eta} \right) = f \]

(24)

3. Numerical Method

We propose a A Finite Element - Characteristic - Finite Difference method. Given a time step \( \Delta t \), and a interval length \( \Delta z \) in the vertical direction, let \( A^n_l \) be a Finite Element approximation of the operator \(-\nabla (k_{xy} \nabla)\) in the layer \( z_l \) at time \( t_n \) and we denote \( u^n_l \) the solution in this level at this time. We use a two-dimensional Finite Element method in each level combined with the Characteristic method. In the vertical direction we approximate the convective term with an upwind first order scheme and the diffusive term with a second order Finite Differences scheme. The crossed derivatives are approximated using prisms with triangular section, which is equivalent to use triangular Finite Element in the horizontal direction and finite difference in the vertical direction. Let \( \omega \) be any of the horizontal sections of the cuboid. We denote \( B_l \) and \( C_l \) the matrix defined by the terms

\[(B_l)_{ij} = \int_{\omega} k_{xy} \left( \frac{\partial h}{\partial x} \frac{\partial \phi_j}{\partial x} + \frac{\partial h}{\partial y} \frac{\partial \phi_j}{\partial y} \right) \phi_i \]

(25)
\[(C_i)_{ij} = \int_\Omega k_{xy} \left( \frac{\partial h}{\partial x} \frac{\partial \varphi_i}{\partial x} + \frac{\partial h}{\partial y} \frac{\partial \varphi_i}{\partial y} \right) \varphi_j \]  

\section{3.1. A one step Euler implicit method}

Given a time step \(\Delta t\), and a interval length \(\Delta z\) in the vertical direction, for each layer \(l = 1, \ldots, L\)

\[
\frac{u_{l}^{n+1} - \bar{u}_{l}^{n}}{\Delta t} + \frac{W_l^+}{\Delta z} (u_{l}^{n+1} - u_{l-1}^{n+1}) - \frac{W_l^-}{\Delta z} (u_{l+1}^{n+1} - u_{l}^{n+1}) \\
+ A_l u_{l+1}^{n+1} + k_2 \frac{-u_{l-1}^{n+1} + 2u_{l}^{n+1} - u_{l+1}^{n+1}}{(\Delta z)^2} \\
+ \frac{1}{2\Delta z} B_l (u_{l+1}^{n+1} - u_{l-1}^{n+1}) + \frac{1}{2\Delta z} C_l (u_{l+1}^{n+1} - u_{l}^{n+1}) \\
\{ i f (l = 1) + \lambda u_{l}^{n+1} = f_{n+1} \} = 0
\]

where \(W_l^+ = \max \{0, W_l\}, W_l^- = \max \{0, -W_l\}\) and \(\bar{u}_{l}^{n}\) is given by \(\bar{u}_{l}^{n} = u_{l}^{n} \circ X^n\)

where \(X^n(x) = X(x; t^n)\) is the solution at time \(t_n\) of the final value problem

\[
\frac{dX}{dt} = V \\
X(x, t^{n+1}) = x
\]

\section{3.2. Splitting Method}

In the former scheme, all the levels are coupled, so is not very suitable from a practical point of view. Now we consider a splitting method where the problem can be solved at each level separately.

For \(l = 1, \ldots, L\)

\[
\frac{u_{l}^{n+1/4} - \bar{u}_{l}^{n}}{\Delta t/2} = 0
\]

\[
\frac{u_{l}^{n+1/2} - u_{l}^{n+1/4}}{\Delta t} + \frac{W_l^+}{\Delta z} (u_{l}^{n+1/2} - u_{l-1}^{n+1/2}) \\
+ \frac{1}{2} A_l u_{l}^{n+1/2} + k_2 \frac{-u_{l-1}^{n+1/2} + u_{l}^{n+1/2}}{(\Delta z)^2} \\
+ \frac{1}{2\Delta z} (-B_l u_{l}^{n+1/2}) + \frac{1}{2\Delta z} (-C_l u_{l}^{n+1/2}) \\
\{ i f (l = 1) + \lambda u_{l}^{n+1/2} = \frac{1}{2} f_{n+1/2} \} = 0
\]
For $l = L, ..., 1$

\[
\begin{align*}
\frac{u_t^{n+3/4} - u_t^{n+1/2}}{\Delta t} &= - \frac{W_l^-}{\Delta z} (u_{t+1}^{n+3/4} - u_t^{n+3/4}) \\
&+ \frac{1}{2} A_l u_t^{n+3/4} + k_z u_t^{n+3/4} - u_t^{n+3/4} \\
&+ \frac{1}{2\Delta z} (B_l u_{t+1}^{n+3/4}) + \frac{1}{2\Delta z} (C_l u_{t+1}^{n+3/4}) \\
&\{i f (l = 1) + \lambda u_t^{n+3/4} \} = \frac{1}{2} f_{n+3/4} \quad (32)
\end{align*}
\]

\[
\frac{u_t^{n+1} - u_t^{n+3/4}}{\Delta t/2} = 0 \quad (33)
\]

**Remark** The term $(l = 1) \lambda u_t^{n+1}$ represents the eventual absorption by the terrain and it appears only in the surface level $(l = 1)$.

### 3.3. Justification of the splitting method

In order to simplify we consider the former problem without convective terms and crossed derivatives. The discrete equations can be written using matricial notation,

\[
d\frac{du}{dt} + Au + Tu = f \quad (34)
\]

where

\[
u = (u_1, \cdots, u_t, \cdots, u_L)^t
\]

\[
A = \begin{bmatrix}
A_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & A_L
\end{bmatrix}
\]

\[
T = \frac{1}{(\Delta z)^2} \begin{pmatrix}
2 & -1 & 0 & \cdots & 0 \\
-1 & 2 & -1 & \cdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & -1 & 2
\end{pmatrix}
\]

\[
\text{(35)}
\]

We split the tridiagonal matrix $T, T = U + L$ where
\[ L = \frac{1}{(\Delta z)^2} \left( \begin{array}{cccccc} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -1 & 1 \end{array} \right) \] (36)

and

\[ U = \frac{1}{(\Delta z)^2} \left( \begin{array}{cccccc} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 1 \end{array} \right) \] (37)

Finally the splitting algorithm is

\[
\begin{align*}
\frac{u^{n+1/2} - u^n}{\Delta t} + \frac{1}{2} A u^{n+1/2} + L u^{n+1/2} &= \frac{1}{2} f^{n+1/2} \\
\frac{u^{n+1} - u^{n+1/2}}{\Delta t} + \frac{1}{2} A u^{n+1} + U u^{n+1} &= \frac{1}{2} f^{n+1}
\end{align*}
\] (38) (39)

4. Parallel implementation

If we observe equations 30-33, the horizontal convective and diffusive terms can be computed simultaneously at all levels, but not the vertical convective and diffusive terms, and the crossed derivatives. We propose a further splitting in order to allow parallel computation.

The new splitting parallel algorithm is:

- For \( l = 1, \ldots, L \)
  - Parallel Loop

\[
\begin{align*}
\frac{u_{l}^{n+1/6} - \bar{u}_{l}^{n}}{\Delta t/2} &= 0 \\
\frac{u_{l}^{n+1/3} - u_{l}^{n+1/6}}{\Delta t} + \frac{1}{2} A u_{l}^{n+1/3} + \frac{1}{2} A u_{l}^{n+1/3} &= \frac{1}{2} f^{n+1/3}
\end{align*}
\]
• Non-parallel loop
\[ \frac{u_i^{n+1/2} - u_i^{n+1/3}}{\Delta t} + \frac{W_t^+(u_i^{n+1/2} - u_{i-1}^{n+1/2})}{\Delta z} + k_z \frac{-u_{i-1}^{n+1/2} + u_i^{n+1/2}}{(\Delta z)^2} + \frac{1}{2\Delta z}(-B_t u_{i-1}^{n+1/2}) + \frac{1}{2\Delta z}(-C_t u_{i-1}^{n+1/2}) = 0 \]

For \( l = L, \ldots, 1 \)

• Non-parallel Loop
\[ \frac{u_i^{n+2/3} - u_i^{n+1/2}}{\Delta t} - \frac{W_t^-(u_{i+1}^{n+2/3} - u_i^{n+2/3})}{\Delta z} + k_z \frac{u_i^{n+2/3} - u_{i+1}^{n+2/3}}{(\Delta z)^2} + \frac{1}{2\Delta z}(B_t u_{i+1}^{n+2/3}) + \frac{1}{2\Delta z}(C_t u_{i+1}^{n+2/3}) = 0 \]

• Parallel loop
\[ \frac{u_i^{n+5/6} - u_i^{n+2/3}}{\Delta t} + \frac{1}{2} A_t u_i^{n+5/6} \quad \{ if (l == 1) \quad + \frac{1}{2} \lambda u_i^{n+5/6} \} = \frac{1}{2} f^{n+5/6} \]
\[ \frac{u_i^{n+1} - u_i^{n+5/6}}{\Delta t/2} = 0 \]

This new parallel algorithm significantly reduces the computation time especially as the number of layers increases, as we can see in the numerical example presented in next section. These examples have been computed on a Dell Precision T7500, with two processors Intel Xeon X5550 and 24 GB RAM. In the following table, we compare computation time of both schemes, the parallel algorithm and the sequential algorithm, for 4, 8, 16 and 24 layers, with a time reduction factor of almost 50 for 24 layers.

5. Numerical Results

The following results correspond to a simulation in an area of 5 \times 5 km² in Cofrentes (a small town near Valencia, Spain) with a maximum height of 213 km. The air layer studied is about 1 km high (see Figure 1).
Cuadro 1: Run times up to 24 layers. Real Time: 1 Hour

<table>
<thead>
<tr>
<th>threads/layers</th>
<th>Run Time (s)</th>
<th>Sequential Time (s)</th>
<th>Red. factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>37</td>
<td>306</td>
<td>8.27</td>
</tr>
<tr>
<td>8</td>
<td>61</td>
<td>856</td>
<td>14.03</td>
</tr>
<tr>
<td>16</td>
<td>94</td>
<td>2867</td>
<td>30.05</td>
</tr>
<tr>
<td>24</td>
<td>106</td>
<td>5260</td>
<td>49.62</td>
</tr>
</tbody>
</table>

Figura 1: Air layer and surface

The wind field at ground level is shown in Figure 2. The wind model and its numerical solution is described in [3], [4] and [5].

Figura 2: velocity field

We assume that at a given time a certain amount of pollutant is released to the atmosphere taking place at the ground surface according to the following
expression (gaussian emission)

\[ f(t, x) = ae^{-\left(\frac{\ln(2)}{c}t \right)} e^{-\left((X[0]-x[0])^2+(X[1]-x[1])^2/(2b^2)\right)} \]

where

- \( t \) is the time in seconds
- \( a = 100 \) pre-exponential factor.
- \( b = 100 \) is the standard deviation of the gaussian distribution.
- \( c = 300 \) is the half life time of the pollutant emission in seconds.
- \( X = [500, 4500] \) is the point where the pollutant emission takes place.

The other physical values are

- Horizontal Diffusion \( k_{xy} = 10^{-1} \)
- Vertical Diffusion \( k_z = 10^{-3} \)
- Absorption coefficient in the surface level \( \lambda = 0.001 \)

Figure 3 shows the concentration at first layer: at initial time, after 10, 20 and 30 time steps respectively.

Figure 4 shows the concentration at third layer: after 10, 20, 30 and 40 time steps respectively.

Figure 5 shows the concentration at fifth layer: after 10, 20, 30 and 40 time steps respectively.
Figura 3: Concentration in the first level at different time steps
Figura 4: Concentration in the third level at different time steps
Figura 5: Concentration in the fifth level at different time steps

Acknowledgement: Supported by the research projects:

Bibliografía


