Near Field Atmospheric Dispersion Modelling on an Industrial Site Using Combination of Cellular Automata and Artificial Neural Networks

Pierre Lauret, Ph.D. Student, 2nd year
pierre.lauret@mines-ales.fr

Frédéric Heymes\textsuperscript{a}, Laurent Aprin\textsuperscript{a}, Anne Johannet\textsuperscript{a}, Gilles Dusserre\textsuperscript{a}, Laurent Munier\textsuperscript{b}, Emmanuel Lapébie\textsuperscript{b}

\textsuperscript{a}: Ales School of Mines, France
\textsuperscript{b}: French Alternative Energies and Atomic Energy Commission
Summary

- Context of the study
- Machine learning tools
- Methodology
- Results
- Improvements
- Conclusion
Summary

▼ Context of the study
   ► Industrial site
   ► Existing models

► Machine learning tools
► Methodology
► Results
► Improvements
► Conclusion
Summary

- Context of the study
- Machine learning tools
  - Cellular Automata - CA
  - Artificial Neural Networks - ANN
  - Combination
- Methodology
- Results
- Improvements
- Conclusion
Summary

- Context of the study
- Machine learning tools
- Methodology
  - CFD case
  - Database creation
  - Scalar transport equation discretisation
  - ANN - Learning
  - ANN – Optimisation
  - CA-ANN – using the model
- Results
- Improvements
- Conclusion
Summary

- Context of the study
- Machine learning tools
- Methodology
- Results
  - Test cases
  - Discussion
- Improvements
- Conclusion
Summary

- Context of the study
- Machine learning tools
- Methodology
- Results
- Improvements
- Conclusion
Near Field Atmospheric Dispersion Modelling - Cellular Automata and Artificial Neural Networks

<table>
<thead>
<tr>
<th>Context</th>
<th>Machine learning</th>
<th>Methodology</th>
<th>Results</th>
<th>Improvements</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Industrial Site – Flammable/Toxic material storage - Dispersion**

Leakage accident

Petrochemical site, Martigues, France

Impact distance < 1 000 m

Exposure time < 1 h
Existing models / Developed model

From quickness to completeness

► Gaussian models (Solving the Advection-Diffusion Equation)
► Integral models
► Computational Fluid Dynamics (CFD) models
  ► Reynolds Averaged Navier-Stockes equations (RANS)
  ► Large Eddy Simulation (LES)
  ► Direct Numerical Simulation (DNS)

CA-ANN

► Methodology development
► Experimental validation
► Uncertainty Quantification and Limits of used
► Aim: Providing operational model with several goals:

- Quickness
- Completeness
- Near field
- Short time dispersion
- Consideration of obstacles
- Real experiments designed
- Developed model

From quickness to completeness
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

**Transition rules example**

<table>
<thead>
<tr>
<th>Time</th>
<th>State 0</th>
<th>State 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t+dt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t+dt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Cellular Automata evolution**

<table>
<thead>
<tr>
<th>Time</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Monodimensional Cellular Automata Example**
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

Monodimensional Cellular Automata Example

<table>
<thead>
<tr>
<th>t</th>
<th>t+dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td></td>
</tr>
<tr>
<td>t+d</td>
<td>0</td>
</tr>
<tr>
<td>t+</td>
<td>1</td>
</tr>
<tr>
<td>t+dt</td>
<td>0</td>
</tr>
<tr>
<td>t+dt</td>
<td>1</td>
</tr>
</tbody>
</table>

Transition rules example

Cellular Automata evolution
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

**Transition rules example**

<table>
<thead>
<tr>
<th>t</th>
<th>t+dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>« 0 » States</td>
<td>« 1 » States</td>
</tr>
</tbody>
</table>

**Cellular Automata evolution**

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

**Monodimensional Cellular Automata Example**

<table>
<thead>
<tr>
<th>t</th>
<th>t+dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>« 0 » States</td>
<td>« 1 » States</td>
</tr>
</tbody>
</table>
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

<table>
<thead>
<tr>
<th>Transition rules example</th>
<th>Cellular Automata evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>( t_0 )</td>
</tr>
<tr>
<td>( t + dt )</td>
<td>( t_1 )</td>
</tr>
<tr>
<td></td>
<td>( t_2 )</td>
</tr>
<tr>
<td>« 0 » States</td>
<td>( t_3 )</td>
</tr>
<tr>
<td>( t )</td>
<td>( t_4 )</td>
</tr>
<tr>
<td>( t + dt )</td>
<td>( t_5 )</td>
</tr>
<tr>
<td>« 1 » States</td>
<td></td>
</tr>
</tbody>
</table>

Monodimensional Cellular Automata Example
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules

- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

Transition rules example

<table>
<thead>
<tr>
<th>t</th>
<th>t+dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>« 0 » States</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t</th>
<th>t+dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>« 1 » States</td>
<td></td>
</tr>
</tbody>
</table>

Monodimensional Cellular Automata Example
Cellular Automata (CA) – Spatial and temporal representation

- Cellular Automata are designed by:
  - Regular mesh
  - Finite possible states for each cell
  - Transition rules
- State evolution is done by applying local and deterministic rules, uniformly and synchronously for all cells.

Transition rules example:

- States:
  - "0" States
  - "1" States

Monodimensional Cellular Automata Example
Artificial Neural Networks (ANN) – Non linear phenomenon approximation

- Non-linear statistical data modelling tools
- Learning iterative process
Artificial Neural Networks (ANN) – Non linear phenomenon approximation

- Non-linear statistical data modelling tools
- Learning iterative process
- Using the ANN
Artificial Neural Networks (ANN) – Non linear phenomenon approximation

- Non-linear statistical data modelling tools
- Parameters modification to minimize the ANN error
- Database of the phenomenon required
Cellular Automata Artificial Neural Networks ruled (CA-ANN)

Cellular Automata

Spatial and temporal representation

Artificial Neural Networks

Non linear phenomenon approximation

Navier-Stokes equations emulation

Unsteady cases
Free Field Methane Atmospheric dispersion

Puff evolution
- Sequence: short time release of CH₄, free evolution until exiting the numerical domain
- Each time step, case is saved (Velocity field/ CH₄ Concentration)
- Time step duration is related to Courant number

Intervals:
- Wind velocity: 2-20 m.s⁻¹
- Ejection velocity: 2-20 m.s⁻¹
- CH₄ Mass Fraction: 0.2-1
- Time step: 20

Methane ejection with initial velocity – RANS k-ε realizable model
Formatting data

Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + u_x \frac{\partial C}{\partial x} + u_y \frac{\partial C}{\partial y} = \frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + S_c
\]

For each cell i,j:
Formatting data

Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + U_x \frac{\partial C}{\partial x} + U_y \frac{\partial C}{\partial y} = \frac{\partial^2 C}{\partial x^2} + S_c
\]

For each cell \(i,j\):

\[
U_x |_{i,j}
\]

\[
U_y |_{i,j}
\]
**Formatting data**

Starting from advection-diffusion equation (Navier-Stokes):

For each cell $i,j$:

\[
\begin{align*}
\frac{\partial C}{\partial t} + u_x \frac{\partial C}{\partial x} + u_y \frac{\partial C}{\partial y} &= \lambda \frac{\partial^2 C}{\partial x^2} + S_c \\
C_{i,j}^{t_0} &\rightarrow C_{i,j}^{t_0+dt}
\end{align*}
\]

---

<table>
<thead>
<tr>
<th>Context</th>
<th>Machine learning</th>
<th>Methodology</th>
<th>Results</th>
<th>Improvements</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formatting data

Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + U_x \frac{\partial C}{\partial x} + U_y \frac{\partial C}{\partial y} = \frac{\partial^2 C}{\partial x^2} + S_c
\]

For each cell \(i,j\):

\[
C_{i,j}^t = C_{i,j}^{t-1} + \frac{C_{i,j}^t - C_{i,j}^{t-1}}{\Delta x} + \frac{C_{i,j+1}^t - C_{i,j}^{t-1}}{2 \Delta y}
\]
Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + u_x \frac{\partial C}{\partial x_i} = \lambda \frac{\partial^2 C}{\partial x_i^2} + S_c
\]

For each cell \(i,j\):

\[
C_{i,j}^t = C_{i,j}^{t-1} + \frac{C_{i,j}^{t-1} - C_{i,j}^{t+1}}{2\Delta y} + \frac{C_{i,j+1}^{t-1} - C_{i,j-1}^{t-1}}{2\Delta x} - \Delta t \left( \frac{C_{i-1,j}^t - 2C_{i,j}^t + C_{i+1,j}^t}{\Delta x^2} + \frac{C_{i,j+1}^t - 2C_{i,j}^t + C_{i,j-1}^t}{\Delta y^2} \right)
\]
Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + U_x \frac{\partial C}{\partial x} + U_y \frac{\partial C}{\partial y} = \lambda \frac{\partial^2 C}{\partial x^2} + S_c
\]

For each cell \(i,j\):

\[
\begin{align*}
U_x & |_{i,j} \\
U_y & |_{i,j} \\
C_{i,j}^t & - C_{i-1,j}^t + \frac{C_{i,j+1}^t - C_{i,j-1}^t}{2. \Delta y} \\
C_{i-2,j}^t - 2C_{i-1,j}^t + C_{i,j}^t & + \frac{C_{i,j+1}^t + C_{i,j-1}^t - 2C_{i,j}^t}{\Delta x^2}
\end{align*}
\]
Formatting data

Starting from advection-diffusion equation (Navier-Stokes):

\[
\frac{\partial C}{\partial t} + u_x \frac{\partial C}{\partial x} + u_y \frac{\partial C}{\partial y} = \lambda \frac{\partial^2 C}{\partial x^2} + S_c
\]

For each cell \(i,j:\)

\[
C_{i,j}^{t+1} = C_{i,j}^t - \frac{C_{i,j}^t - C_{i-1,j}^t}{\Delta x} \frac{C_{i,j+1}^t - C_{i,j-1}^t}{2 \Delta y} \frac{C_{i-2,j}^t - 2C_{i-1,j}^t + C_{i,j}^t}{\Delta x^2} + \frac{C_{i,j+1}^t + C_{i,j-1}^t - 2C_{i,j}^t}{\Delta y^2} + U_x |_{i,j} \]

\[
U_y |_{i,j}
\]

\[
C_{i,j}^t
\]

\[
\frac{C_{i,j}^t - C_{i-1,j}^t}{\Delta x} \frac{C_{i,j+1}^t - C_{i,j-1}^t}{2 \Delta y} \frac{C_{i-2,j}^t - 2C_{i-1,j}^t + C_{i,j}^t}{\Delta x^2} + \frac{C_{i,j+1}^t + C_{i,j-1}^t - 2C_{i,j}^t}{\Delta y^2}
\]
Formatting data

Starting from advection-diffusion equation (Navier-Stokes):

\[ \frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \nabla \cdot (D \nabla C) + S_c \]

\[ \begin{align*}
C_{i,j}^{t+1} &= C_{i,j}^t + u_x \Delta x + u_y \Delta y + \frac{\Delta t}{\Delta x^2} (C_{i+1,j}^t - 2C_{i,j}^t + C_{i-1,j}^t) \\
&\quad + \frac{\Delta t}{\Delta y^2} (C_{i,j+1}^t - 2C_{i,j}^t + C_{i,j-1}^t) + \frac{\Delta t}{\Delta x} \left( \frac{\partial C}{\partial x} \right)_{i,j}^t + S_c
\end{align*} \]
Simulating a known case and evaluating the CA-ANN

- $C_0$ initialization from an existing case
Using and evaluating the CA-ANN

- $c_0$ initialization from an existing case

Iterative algorithm of the Cellular Automata Artificial Neural Network ruled (CA-ANN)
Using and evaluating the CA-ANN

- $C_0$ initialization from an existing case
Using and evaluating the CA-ANN

- $C_0$ initialization from an existing case
Using and evaluating the CA-ANN

- \( C_0 \) initialization from an existing case
Using and evaluating the CA-ANN

- $C_0$ initialization from an existing case
Unlearned case simulation and evaluation

**Initialization**: using a CFD case (Initial Mass Fraction: 0.89; Wind velocity: 10.2 m.s\(^{-1}\))

**Mass conservation trough time steps:**

![Graph showing mass evolution over time for CFD simulation and CA-ANN model](image)

- **Mass evolution for CFD simulation and CA-ANN model**
- **Time steps**
- **Total mass**
- **Mass (CA-ANN)**
- **Mass (CFD)**
- **+/− 10% CFD**
Unlearned case simulation and evaluation

Initialization: using a CFD case (Initial Mass Fraction: 0.89; Wind velocity: 10.2 m.s⁻¹)

Mass conservation trough time steps

Uncertainty visualisation: Model Vs CFD reality

![Graph showing model concentrations vs CFD concentrations at time step 1. The graph includes a line indicating $y=x$ and a line indicating ±25% variation. The coefficient of determination $R^2 = 0.99996$.](image)
Unlearned case simulation and evaluation

Initialization: using a CFD case (Initial Mass Fraction: 0.89; Wind velocity: 10.2 m.s⁻¹)

Mass conservation through time steps

Uncertainty visualisation: Model Vs CFD reality

![Graph showing model concentrations vs. CFD concentrations at time step 49](image1)

![Diagram showing absolute error at time step](image2)
Unlearned case simulation and evaluation

Initialization: using a CFD case (Initial Mass Fraction: 0.89; Wind velocity: 10.2 m.s\(^{-1}\))

Other evaluation criteria mainly used [1993, Kumar], [2004, Chang&Hanna]:

\[
R^2 = 1 - \frac{RSS}{TSS} \quad \text{FAC2: } 0.5 \leq \frac{C_p}{C_0} \leq 2
\]

\[
FB = 2 \times \left( \frac{C_0 - C_p}{C_0 + C_p} \right) \quad \text{NMSE} = \left( \frac{C_0 - C_p}{C_0 \times C_p} \right)^2
\]

\[
MG = \exp \left( \ln C_0 - \ln C_p \right) \quad \text{VG} = \exp \left[ \ln C_0 - \ln C_p \right]^2
\]
Improvements

Wind field determination
- Experimental (wind tunnel) and numerical database (CFD)
- Influence of cylindrical obstacles
- ANN modelisation
- Several obstacles dimensions
- Several inlet velocity

Coupling wind field model (ANN) and atmospheric dispersion model (CA-ANN)

ANSYS FLUENT 14

Wind tunnel EMA (d = 10 mm; v = 3 m/s⁻¹)
Conclusions

- Existing forecasting models are slow but accurate or fast but not enough appropriate.
- A method using Cellular Automata and Artificial Neural Networks is presented to combine fast calculation and accurate solution.
- Comparison with CFD software gives good agreement and faster processing.
- CA-ANN lightly overestimates the higher concentrations.
- A decay in the determination coefficient through time steps is noted.
- Evaluation criteria ($R^2$, FAC2, FB, NMSE, MG, VG) are within acceptable range.
- Computing optimization is required.
- ANN training optimization process is in progress.
Near Field Atmospheric Dispersion Modelling on an Industrial Site Using Combination of Cellular Automata and Artificial Neural Networks

Contact: pierre.lauret@mines-ales.fr