MEGA
MÉCANIQUE
ENERGÉTIQUE
GÉNIE CIVIL
ACOUSTIQUE

ÉCOLE DOCTORALE 162

Inverse modelling for source characterization in complex industrial sites: Development of the adjoint state method applied to a Lagrangian stochastic dispersion model

Salles Loustau Jean

H22-065
June 13th, 2024

Thesis supervisors: SOULHAC Lionel, INSA Lyon/ECL/Lyon 1/CNRS (LMFA)
EMMANUELLI Ariane, ECL (LMFA)

Co-direction TotalEnergies: DUCLAUX Olivier, TotalEnergies

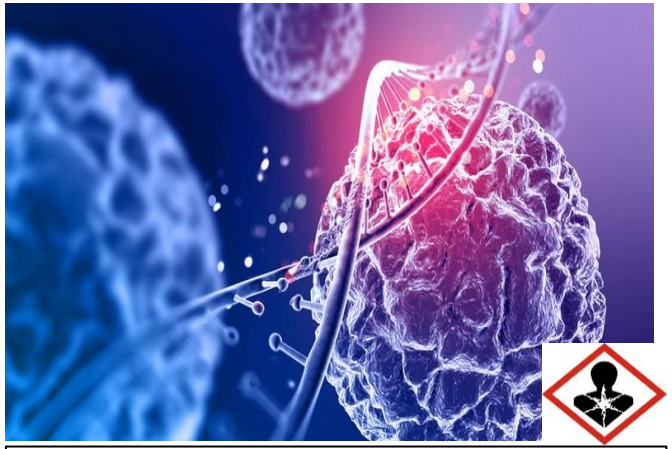
Advisor: NAMMOUR Rami, TotalEnergies

ÉCOLE CENTRALE LYON · Lyon 1 · INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON · cnrs · LMFA · TotalEnergies

INTRODUCTION

ATMOSPHERIC POLLUTION

- **Health risks (cancers, …)**



Cancer cell

- **Environmental risks (global warming, contamination, …)**

Water pollution contamination with mercury, New Idria, USA, 2004

Nuclear radioactive contamination risk

- **Security risks (flammability, explosion, …)**

Refinery fire, Indonesia, 2021 (©AGUS SIPUR / AFP)

- **Need for atmospheric pollutant source identification & quantification in complex industrial sites**

Real-time constraint
Important for industrial companies !

⇨ To improve on-site safety & to evaluate environmental impact

INTRODUCTION

HOW TO QUANTIFY EMISSIONS ?

- Direct measurements of pollutant in atmosphere

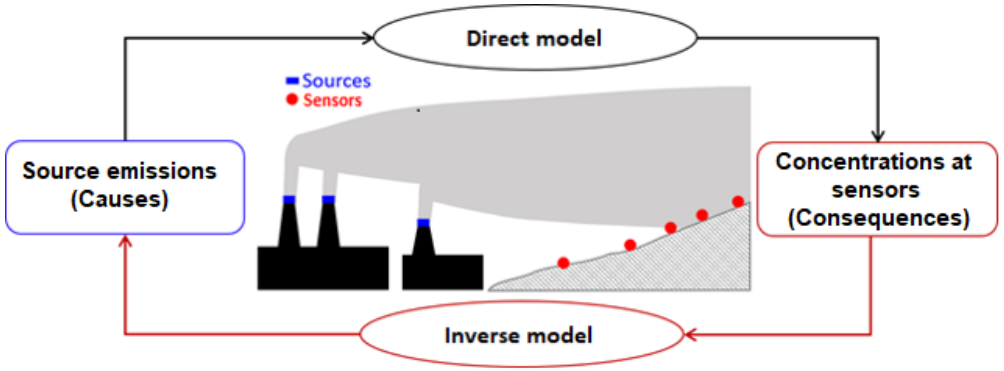
Process: **Source** ⇨ Pollutant transport ⇨ Sensor

Drone AUSEA on a TotalEnergies site (©TotalEnergies)

✗ Leaks & diffuse emissions difficult to characterize, especially in real-time

- **Inverse modelling**: Dispersion prediction & recovery of source characteristics

Process: **Source** ⇦ Pollutant transport ⇦ Sensor



✓ Quality & accuracy of results

- Different approaches exist for inverse modelling:
  - Grid search “brute force” methods (*Ben Salem et al., 2014*)
  - **Minimization problem** (*Gill et al., 1981*)

    Requires cost function Jacobian matrix computation

INTRODUCTION

HOW TO QUANTIFY EMISSIONS ?

- Direct measurements of pollutant in atmosphere

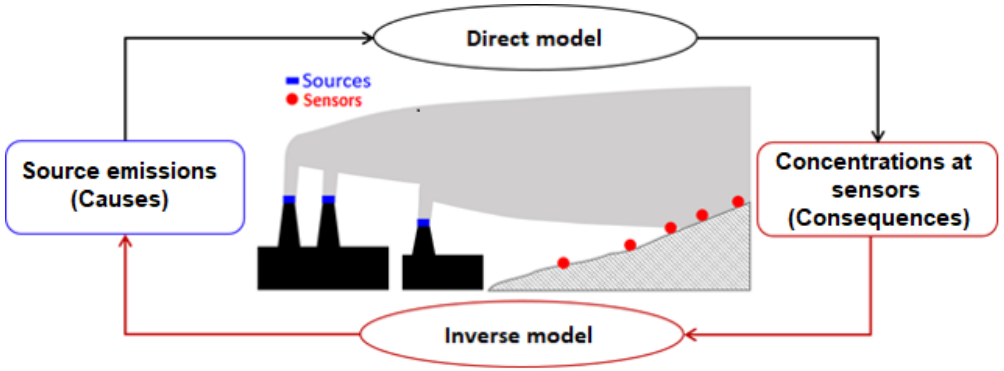
Process: **Source** ⇨ Pollutant transport ⇨ Sensor

Drone AUSEA on a TotalEnergies site (©TotalEnergies)

✗ Leaks & diffuse emissions difficult to characterize, especially in real-time

- **Inverse modelling**: Dispersion prediction & recovery of source characteristics

Process: **Source** ⇦ Pollutant transport ⇦ Sensor



✓ Quality & accuracy of results

- Different approaches exist for inverse modelling:
  - Grid search "brute force" methods (*Ben Salem et al., 2014*)
  - **Minimization problem** (*Gill et al., 1981*)
    Requires cost function Jacobian matrix computation

Time-consuming techniques !
Need to find an alternative:
<u>the adjoint state method</u> (*Chavent, 1974*)

OUTLINE

- Adjoint state method for inverse problem solving
- Application to a Lagrangian stochastic particle dispersion model
- Application results on a numerical test case
- Conclusion

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Forward model definition

- **State equation** - generally implicit (*Plessix, 2006*):

$$\underbrace{F_s}_{n_F \times 1}\left(\underbrace{u_s}_{n_u \times 1}, \underbrace{m_s}_{n_m \times 1}\right) = \underbrace{0}_{n_F \times 1}$$

F_s numerical forward model
m_s model parameters
u_s state (output) variable

- If an explicit relationship f_s exists between u_s and m_s, F_s:

$$F_s(u_s, m_s) = u_s - f_s(m_s) = 0$$

- In atmospheric dispersion context:
  - F_s forward transport and dispersion model of pollutants, originating from a source s
  - u_s concentration vector
  - m_s source parameter vector

- m_s: **Source position** $x_s(x_s, y_s, z_s)$ & **flow rate** $q_s \Rightarrow n_m = 4$

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Methodology for inverse problem solving

- No explicit analytical inversion of the state equation generally !

- Inverse problem must be considered as a **minimization problem**

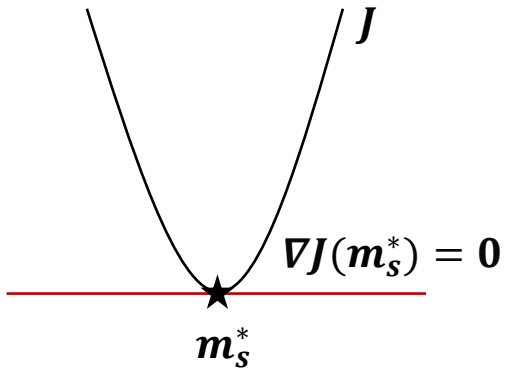
⇨ To minimize the **<u>difference</u>** between **the observation data d** & **the model output data u_s** provided by F_s

1. Define a cost function J depending on model parameters m_s to optimize

$$\underbrace{J}_{1\times 1}\ (\underbrace{m_s}_{n_m\times 1}) = \underbrace{E}_{1\times 1}\ (\underbrace{u_s}_{n_u\times 1}\ ,\ \underbrace{m_s}_{n_m\times 1})$$

with E the error functional of the difference

2. Compute ∇J as $m_s^* = \min_{m_s} J(m_s) \Longrightarrow \nabla J(m_s^*) = 0$



3. Solving of the minimization problem, iteratively, to recover m_s

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Methodology for inverse problem solving

- No explicit analytical inversion of the state equation generally !

- Inverse problem must be considered as a **minimization problem**

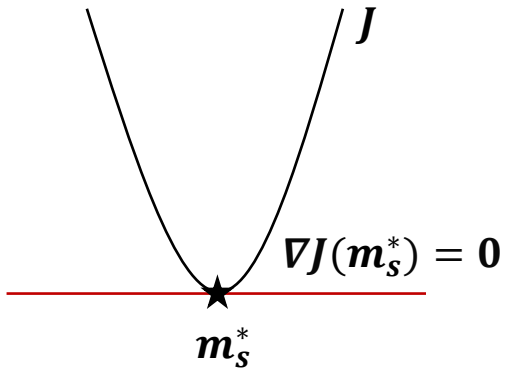
⇨ To minimize the **<u>difference</u>** between **the observation data d** & **the model output data u_s** provided by F_s

1. Define a cost function J depending on model parameters m_s to optimize

$$\underbrace{J}_{1\times 1}\ (\underbrace{m_s}_{n_m\times 1}) = \underbrace{E}_{1\times 1}\ (\underbrace{u_s}_{n_u\times 1}\ ,\ \underbrace{m_s}_{n_m\times 1})$$

with E the error functional of the difference

2. Compute ∇J as $m_s^* = \min_{m_s} J(m_s) \Longrightarrow \nabla J(m_s^*) = 0$



3. Solving of the minimization problem, iteratively, to recover m_s

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Introduction of the adjoint state equation to compute the gradient

- Classically, calculating $\boldsymbol{\nabla J}$ requires the computation of the whole Jacobian matrix

$$\underbrace{\boldsymbol{\nabla J}}_{\boldsymbol{n_m \times 1}} := \left(\frac{\boldsymbol{dJ}}{\boldsymbol{dm_s}}\right)^{\boldsymbol{T}} = \left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial u_s}}\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}} + \frac{\boldsymbol{\partial E}}{\boldsymbol{\partial m_s}}\right)^{\boldsymbol{T}} = \underbrace{\left(\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_m \times n_u}} \underbrace{\left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial u_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_u \times 1}} + \underbrace{\left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial m_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_m \times 1}}$$

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Introduction of the adjoint state equation to compute the gradient

- Classically, calculating $\boldsymbol{\nabla J}$ requires the computation of the whole Jacobian matrix

$$\underbrace{\boldsymbol{\nabla J}}_{\boldsymbol{n_m \times 1}} := \left(\frac{\boldsymbol{dJ}}{\boldsymbol{dm_s}}\right)^{\boldsymbol{T}} = \left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial u_s}}\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}} + \frac{\boldsymbol{\partial E}}{\boldsymbol{\partial m_s}}\right)^{\boldsymbol{T}} = \underbrace{\left(\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_m \times n_u}} \underbrace{\left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial u_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_u \times 1}} + \underbrace{\left(\frac{\boldsymbol{\partial E}}{\boldsymbol{\partial m_s}}\right)^{\boldsymbol{T}}}_{\boldsymbol{n_m \times 1}}$$

- Matrix $\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}}$ is the bottleneck term
  - $\boldsymbol{u_s}$ does not depend generally explicitly on $\boldsymbol{m_s}$ ⇨ **Impossible to differentiate $\boldsymbol{u_s}$!**
  - $\frac{\boldsymbol{du_s}}{\boldsymbol{dm_s}}$ must be computed for each perturbation $\boldsymbol{dm_s}$, typically on each grid point.
    ⇨ **Can exceed many thousands at industrial scale, leading to large computation cost !**

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Introduction of the adjoint state equation to compute the gradient

- Classically, calculating ∇J requires the computation of the whole Jacobian matrix

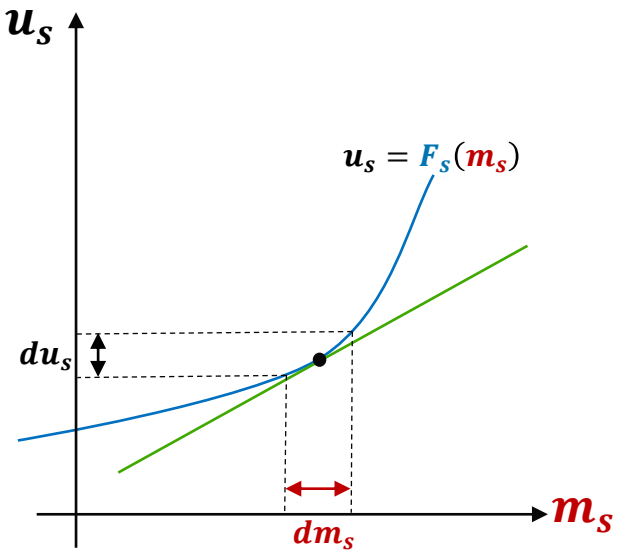
$$\underbrace{\nabla J}_{n_m \times 1} := \left(\frac{dJ}{dm_s}\right)^T = \left(\frac{\partial E}{\partial u_s}\frac{du_s}{dm_s} + \frac{\partial E}{\partial m_s}\right)^T = \underbrace{\left(\frac{du_s}{dm_s}\right)^T}_{n_m \times n_u}\underbrace{\left(\frac{\partial E}{\partial u_s}\right)^T}_{n_u \times 1} + \underbrace{\left(\frac{\partial E}{\partial m_s}\right)^T}_{n_m \times 1}$$

- **Adjoint state method**: alternate way to efficiently compute ∇J

  - Formally derives adjoint equations from transport models (*Pudykiewicz, 1998*) providing sensitivity of model output u_s to input variables m_s

  - Instead of computing $\frac{du_s}{dm_s}$, solving of a linear system, **the adjoint state equation**, specifying the adjoint state λ_s:

$$\underbrace{\left(\frac{\partial F_s}{\partial u_s}\right)^T}_{n_u \times n_F}\underbrace{\lambda_s}_{n_F \times 1} = \underbrace{\left(\frac{\partial E}{\partial u_s}\right)^T}_{n_u \times 1}$$



ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Introduction of the adjoint state equation to compute the gradient

- Classically, calculating ∇J requires the computation of the whole Jacobian matrix

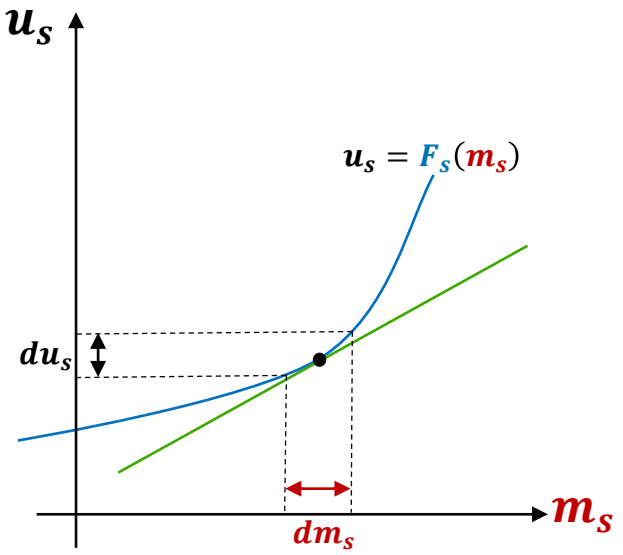
$$\underbrace{\nabla J}_{n_m \times 1} := \left(\frac{dJ}{dm_s}\right)^T = \left(\frac{\partial E}{\partial u_s}\frac{du_s}{dm_s} + \frac{\partial E}{\partial m_s}\right)^T = \underbrace{\left(\frac{du_s}{dm_s}\right)^T}_{n_m \times n_u}\underbrace{\left(\frac{\partial E}{\partial u_s}\right)^T}_{n_u \times 1} + \underbrace{\left(\frac{\partial E}{\partial m_s}\right)^T}_{n_m \times 1}$$

- **Adjoint state method**: alternate way to efficiently compute ∇J

  - Formally derives adjoint equations from transport models (*Pudykiewicz, 1998*) providing sensitivity of model output u_s to input variables m_s

  - Instead of computing $\frac{du_s}{dm_s}$, solving of a linear system, **the adjoint state equation**, specifying the adjoint state λ_s:

$$\underbrace{\left(\frac{\partial F_s}{\partial u_s}\right)^T}_{n_u \times n_F}\underbrace{\lambda_s}_{n_F \times 1} = \underbrace{\left(\frac{\partial E}{\partial u_s}\right)^T}_{n_u \times 1}$$



- ∇J becomes:

$$\underbrace{\nabla J}_{n_m \times 1} := \left(\frac{\mathrm{d}J}{\mathrm{d}m_s}\right)^T = -\underbrace{\left(\frac{\partial F_s}{\partial m_s}\right)^T}_{n_m \times n_F}\underbrace{\lambda_s}_{n_F \times 1} + \underbrace{\left(\frac{\partial E}{\partial m_s}\right)^T}_{n_m \times 1}$$

- **λ_s is a vector, independent of the optimization parameter number, reducing computation time !**

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Extension to a Markovian explicit iterative model with least-square misfit

- **Adjoint method** widely used in atmospheric dispersion field

  - Applications to Gaussian and Eulerian models (*Pudykiewicz, 1998; Giering, 2000*)

  - Not known application to forward Lagrangian Stochastic (LS) models
    BUT:
    - ✓ Suitable for modelling turbulent dispersion in complex environment
    - ✓ Reasonable computational cost

    ⇨ **Objective:** Application to a Lagrangian Stochastic model

- **Case definition**:
  Explicit
  Iterative
  Markov process
  Least-square error functional

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Extension to a Markovian explicit iterative model with least-square misfit

- Under these assumptions + observations only at final N^{th} iteration, **linear adjoint system** becomes iterative
- Computation of ∇J, using λ_s^1 found solving iteratively the **adjoint state equation**

$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m \times n_1} \underbrace{\prod_{j=1}^{N-1}\left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1 \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

ADJOINT STATE METHOD FOR INVERSE PROBLEM SOLVING
Extension to a Markovian explicit iterative model with least-square misfit

- Under these assumptions + observations only at final N^{th} iteration, **linear adjoint system** becomes iterative
- Computation of ∇J, using λ_s^1 found solving iteratively the **adjoint state equation**

$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m \times n_1} \underbrace{\prod_{j=1}^{N-1} \left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1 \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

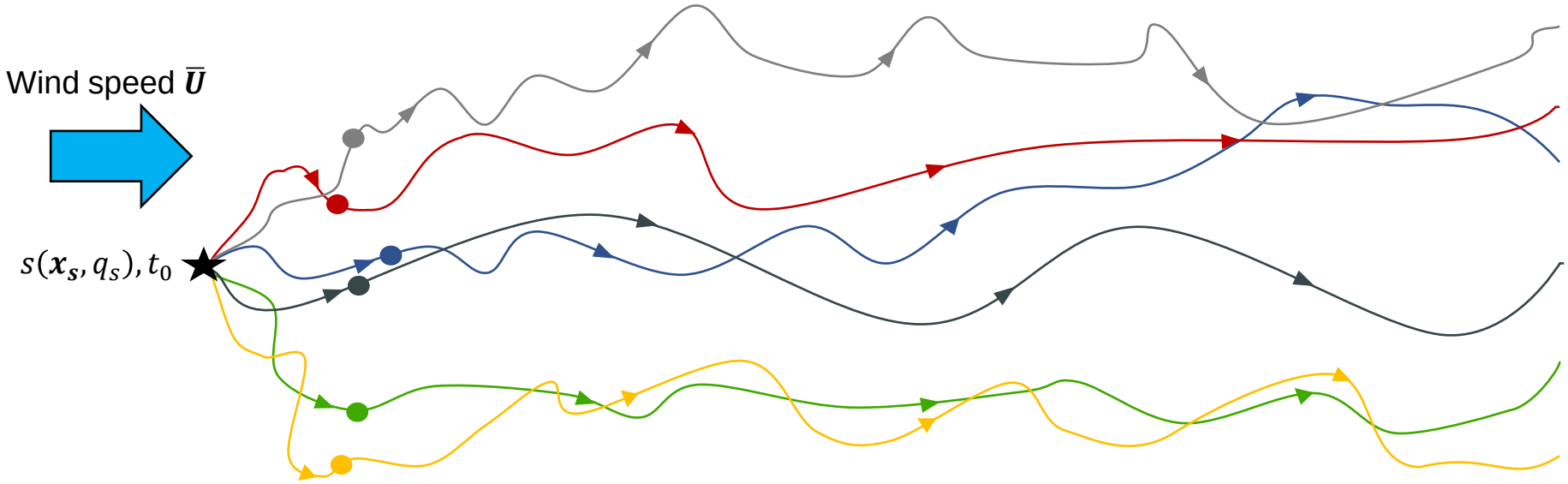
- Efficient way to compute ∇J (chain rule)
- ∇J obtained computing only a product iteratively
- A priori knowledge of u_s^N, d^N & N not necessary

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

- Simplified LS model ⇨ Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stresses

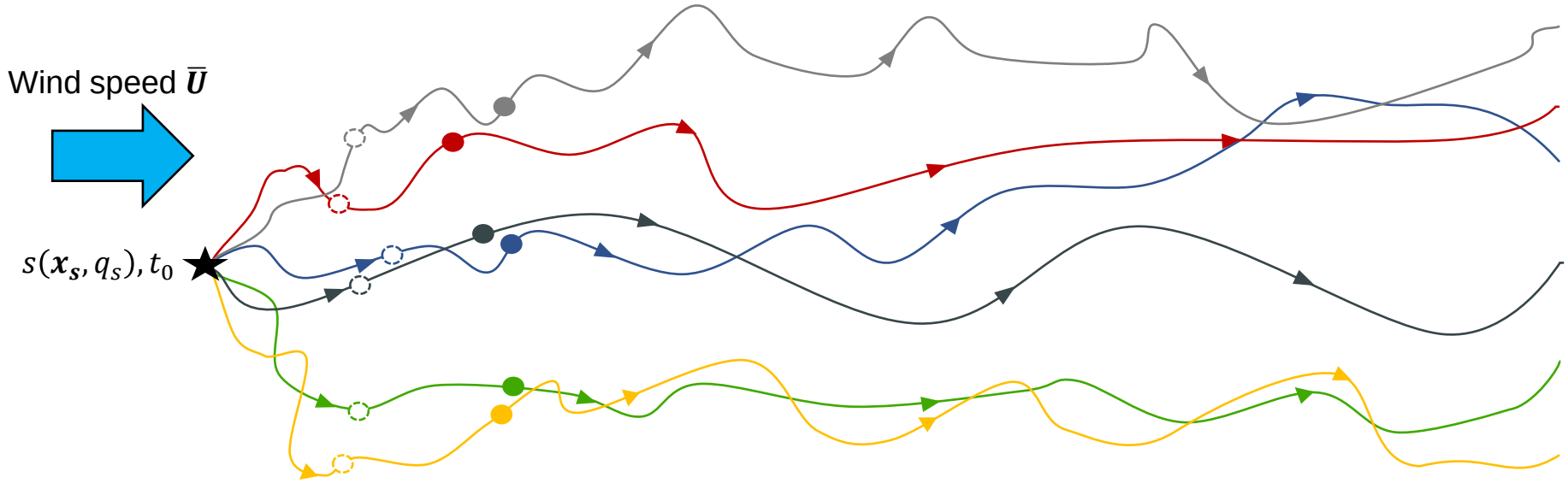
APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

- Simplified LS model ⇨ Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stresses
- **3D/CFD Lagrangian models**: Describes the turbulent pathway simulation of thousands of particles, through a stochastic process



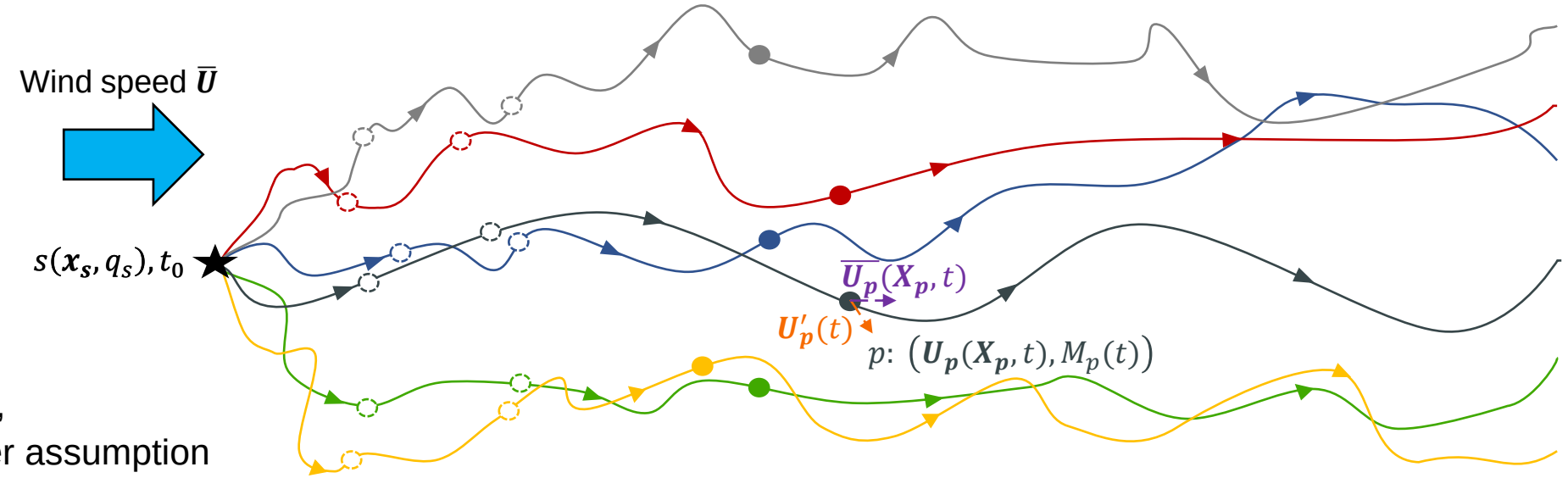
APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

- Simplified LS model ⇨ Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stresses
- **3D/CFD Lagrangian models**: Describes the turbulent pathway simulation of thousands of particles, through a stochastic process



APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

- Simplified LS model ⇨ Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stresses
- **3D/CFD Lagrangian models**: Describes the turbulent pathway simulation of thousands of particles, through a stochastic process
- At each time t, a particle p is fully described by:
  - its **fluctuating velocity** $U'_{p,i}(t)$, for the i^{th} space component
  - its **pseudo-mass** $M_p(t)$, constant over time under assumption of decay process absence

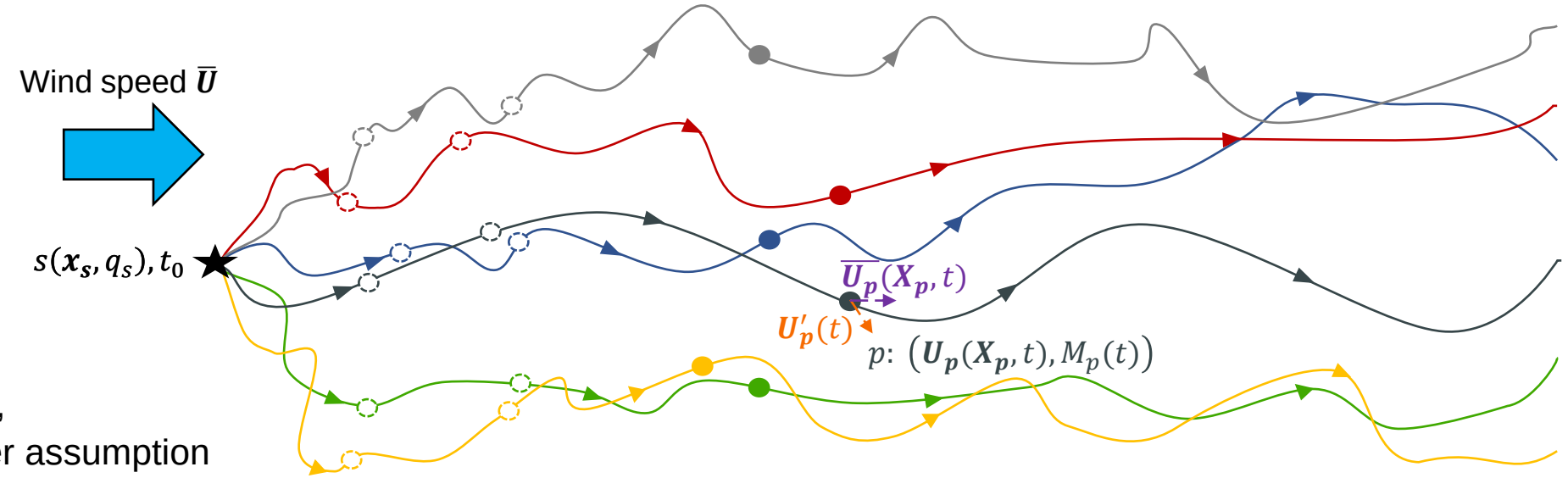


$$\begin{cases} X_{p,i}(t+\delta t) = X_{p,i}(t) + \left(\overline{u_i}(\boldsymbol{X_p}, t) + U'_{p,i}(t)\right)\ \delta t \\ U'_{p,i}(t+\delta t) = U'_{p,i}(t) + \underbrace{\delta U'_{p,i}}_{\text{Stochastic variation}} \end{cases}$$

with $X_{p,i}(t_0) = x_{s,i}$

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

- Simplified LS model ⇨ Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stresses
- **3D/CFD Lagrangian models**: Describes the turbulent pathway simulation of thousands of particles, through a stochastic process
- At each time t, a particle p is fully described by:
  - its **fluctuating velocity** $U'_{p,i}(t)$, for the i^{th} space component
  - its **pseudo-mass** $M_p(t)$, constant over time under assumption of decay process absence



$$\begin{cases} X_{p,i}(t+\delta t) = X_{p,i}(t) + \left(\overline{u_i}(\boldsymbol{X_p}, t) + U'_{p,i}(t)\right)\ \delta t \\ U'_{p,i}(t+\delta t) = U'_{p,i}(t) + \delta U'_{p,i} \end{cases}$$

Stochastic variation

with $X_{p,i}(t_0) = x_{s,i}$

Generalized Langevin Equation[1]

[1]Thomson, D. J. (1987). "Criteria for the selection of stochastic models of particle trajectories in turbulent flows". In: Journal of Fluid Mechanics 180.-1, pp. 529

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Generalized Langevin equation

- Stochastic variation $\delta U'_{\mathrm{p},i}$, for the i^{th} space component, described by the **Generalized Langevin Equation**[1], **for Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stress tensor**:

$$\delta U'_{p,i} = \left[-\frac{1}{T_L} U'_{p,i}\right] \delta t + \sigma_u \sqrt{\frac{2}{T_L}} \sqrt{\delta t} \xi_{p,u_i}$$

  - T_L Lagrangian correlation time of the considered particle
  - σ_u velocity amplitude fluctuation

- **Describes Brownian motion of small dimension particle in a fluid ⇨ Markov process**
- Allows to reproduce some statistics on correlation time between particles

[1]Thomson, D. J. (1987). “Criteria for the selection of stochastic models of particle trajectories in turbulent flows”. In: Journal of Fluid Mechanics 180.-1, pp. 529

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Generalized Langevin equation

- Stochastic variation $\delta U'_{\mathrm{p},i}$, for the i^{th} space component, described by the **Generalized Langevin Equation**[1], **for Gaussian steady isotropic homogeneous turbulence and diagonal Reynolds stress tensor**:

$$\delta U'_{p,i} = \left[-\frac{1}{T_L} U'_{p,i}\right] \delta t + \sigma_u \sqrt{\frac{2}{T_L}} \sqrt{\delta t} \xi_{p,u_i}$$

  - T_L Lagrangian correlation time of the considered particle
  - σ_u velocity amplitude fluctuation

- **Describes Brownian motion of small dimension particle in a fluid ⇨ Markov process**
- Allows to reproduce some statistics on correlation time between particles

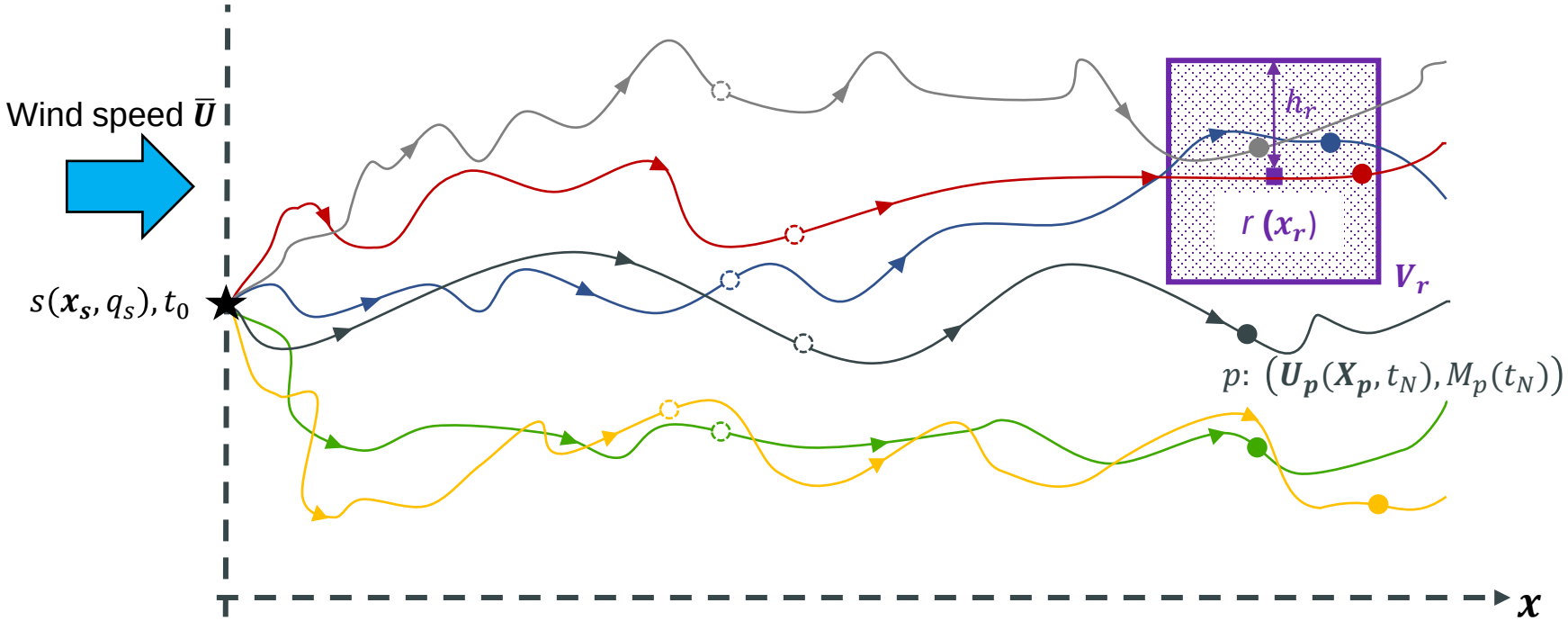
Simplification of the real atmosphere

For the rest of this presentation, work has been done under these assumptions, for understanding of the adjoint state method approach

[1]Thomson, D. J. (1987). “Criteria for the selection of stochastic models of particle trajectories in turbulent flows”. In: Journal of Fluid Mechanics 180.-1, pp. 529

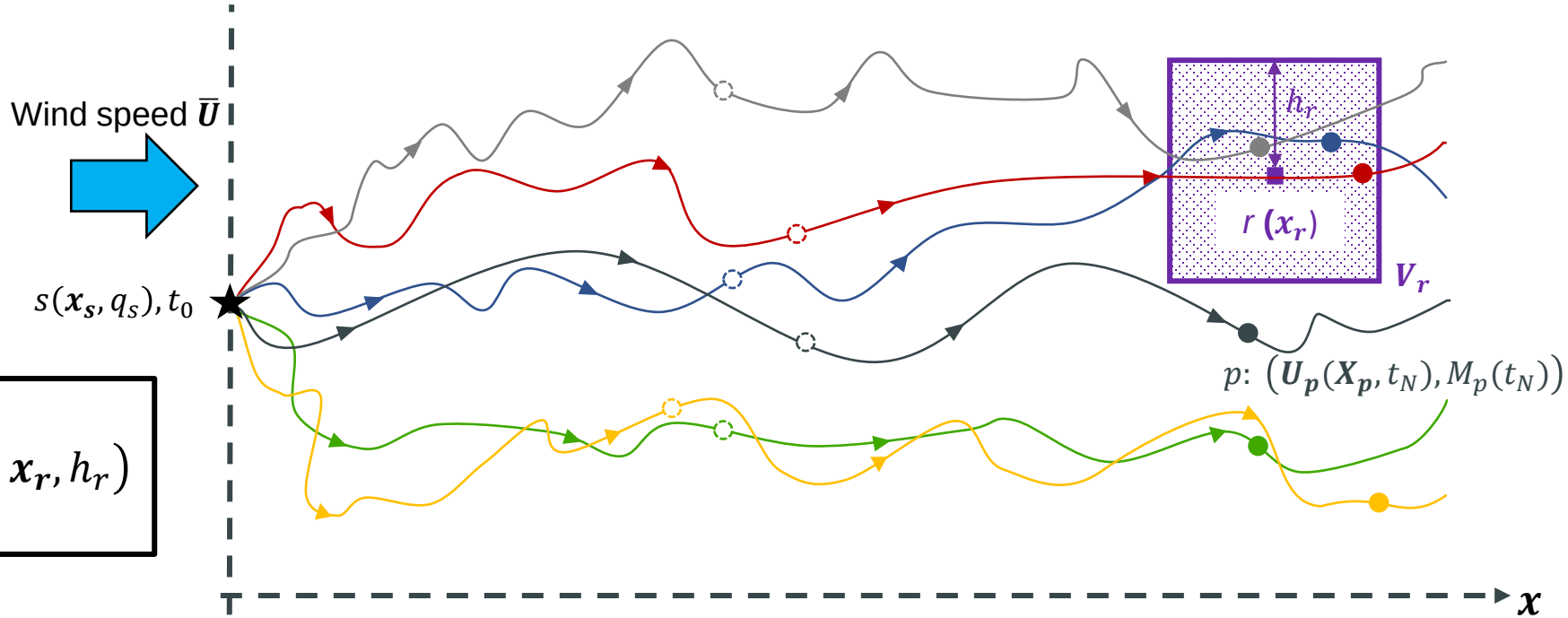
APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Concentration computation

- **Objective**: Compute average concentration $\overline{C_s}(\boldsymbol{x_r}, t_N)$ over volume $\boldsymbol{V_r}$ centred on sensor position $\boldsymbol{x_r}$ for t_N



APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Concentration computation

- **Objective**: Compute average concentration $\overline{C_s}(\boldsymbol{x_r}, t_N)$ over volume $\boldsymbol{V_r}$ centred on sensor position $\boldsymbol{x_r}$ for t_N



- **Density kernel approach**[2]:

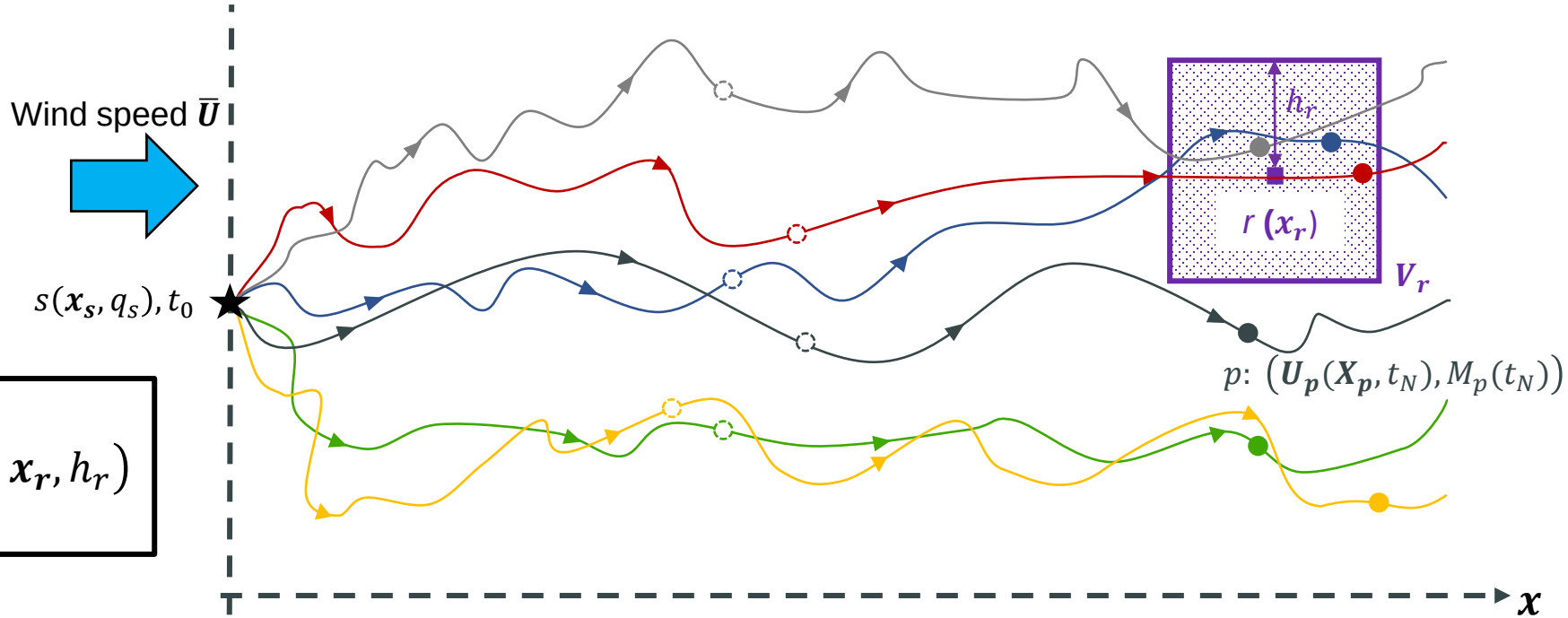
$$\overline{C_s}(\boldsymbol{x_r}, t_N) = \sum_{p=1}^{N_p} M_p(t_N)\, K\left(\boldsymbol{X_p}(t_N) - \boldsymbol{x_r}, h_r\right)$$

- K a kernel function, modelling the detector response function of sensor r
- h_r smoothing radius

[2]Lorimer, G. S. (1986). "The kernel method for air quality modelling-I. Mathematical foundation". In: Atmospheric Environment 20-7, pp. 1447-14520

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Concentration computation

- **Objective**: Compute average concentration $\overline{C_s}(\boldsymbol{x_r}, t_N)$ over volume $\boldsymbol{V_r}$ centred on sensor position $\boldsymbol{x_r}$ for t_N



- **Density kernel approach**[2]:

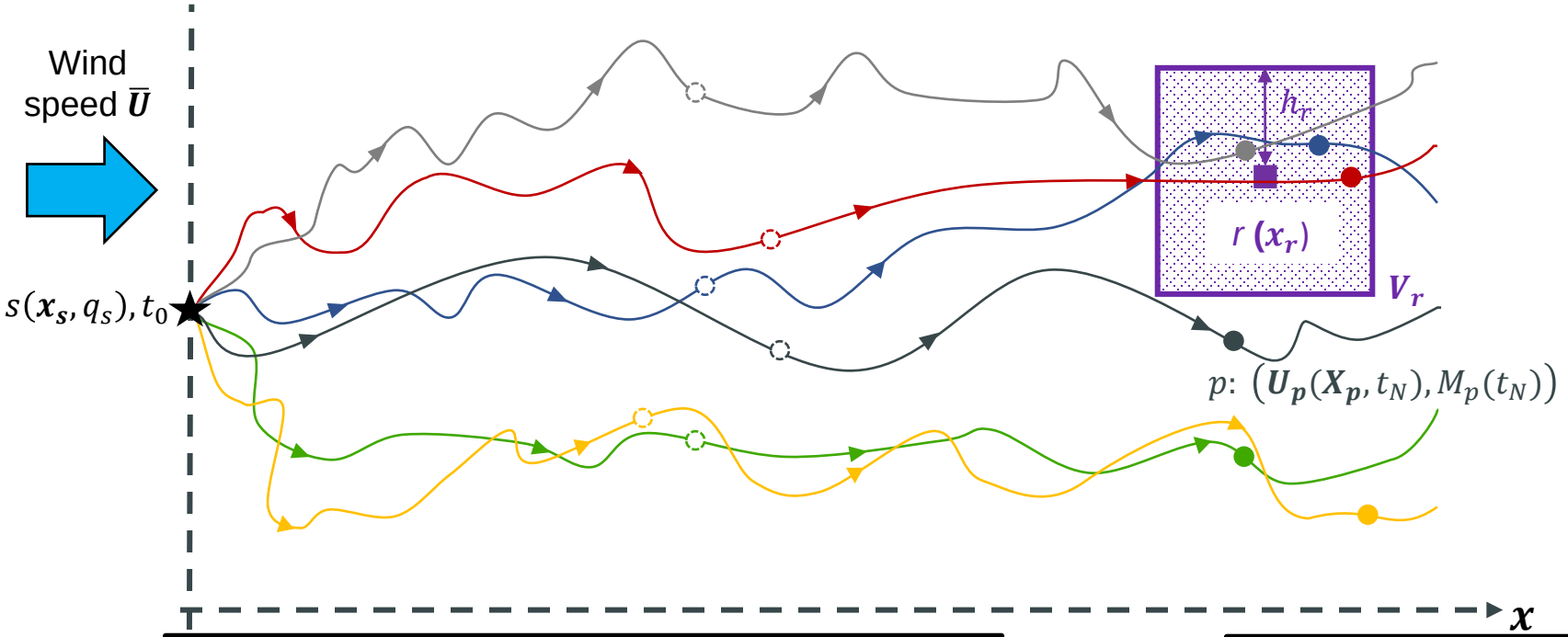
$$\overline{C_s}(\boldsymbol{x_r}, t_N) = \sum_{p=1}^{N_p} M_p(t_N)\, K\left(\boldsymbol{X_p}(t_N) - \boldsymbol{x_r}, h_r\right)$$

  - K a kernel function, modelling the detector response function of sensor r
  - h_r smoothing radius

- $\overline{C_s}$ considered as particle density function, affected by M_p, over $\boldsymbol{V_r}$:
  the density of particles inside $\mathbf{V_r}$ which have been transported forward in time from source s.

[2]Lorimer, G. S. (1986). "The kernel method for air quality modelling-I. Mathematical foundation". In: Atmospheric Environment 20-7, pp. 1447-14520

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL
Adjoint computation



$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m \times n_1} \underbrace{\prod_{j=1}^{N-1} \left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1 \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\sum_{t=1}^{N} \sum_{p=1}^{N_p} O_{p,s}\right)}_{n_m \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

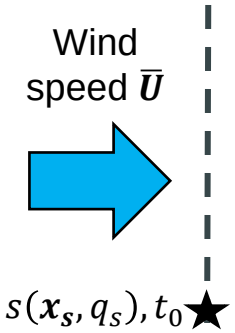
with

$$O_{p,s} = \underbrace{P_{p,s}^T \left[\prod_{j=1}^{N-2} Q_{p,s}^{j+1\,T}\right] R_{p,s}^T}_{n_m \times n_N}$$

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

Adjoint computation-Disaggregation step

Disaggregation: Instant emission of N_p particles from source s, at t_0



$$\underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m \times 1} = \left[\underbrace{P_{1,s}^T}_{n_m \times 7} \cdots \underbrace{P_{p,s}^T}_{n_m \times 7} \cdots \underbrace{P_{N_p,s}^T}_{n_m \times 7}\right]$$

$P_{p,s}^T$ sensitivity matrix of p characteristics (X_p, U'_p and M_p) w.r.t. m_s at t_0

$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m \times n_1} \underbrace{\prod_{j=1}^{N-1}\left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1 \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

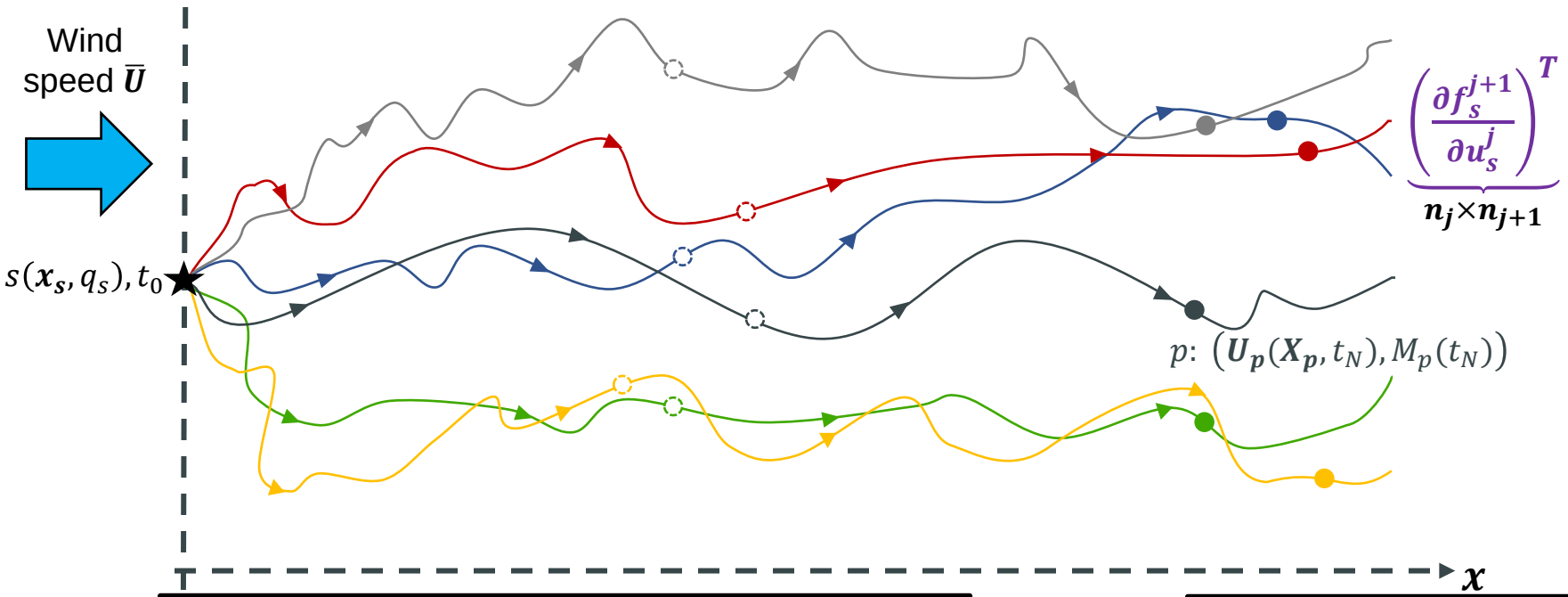
$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\sum_{t=1}^{N}\sum_{p=1}^{N_p} O_{p,s}\right)}_{n_m \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N \times 1}$$

with $$O_{p,s} = \underbrace{P_{p,s}^T \left[\prod_{j=1}^{N-2} Q_{p,s}^{j+1\,T}\right] R_{p,s}^T}_{n_m \times n_N}$$

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

Adjoint computation-Transport step

Transport / Advection: Advection of particles at each time step δt, for N iterations (Lagrangian stochastic path)



$$\underbrace{\left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_j \times n_{j+1}} = diag\left(\underbrace{Q_{1,s}^{j+1\,T}}_{7\times7}, \cdots, \underbrace{Q_{p,s}^{j+1\,T}}_{7\times7}, \cdots, \underbrace{Q_{N_p,s}^{j+1\,T}}_{7\times7}\right)$$

$Q_{p,s}^{j+1\,T}$ sensitivity matrix of p characteristics at $(j+1)^{th}$ iteration w.r.t. j^{th} one

$$\underbrace{\nabla J}_{n_m\times1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m\times n_1} \underbrace{\prod_{j=1}^{N-1}\left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1\times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N\times1}$$

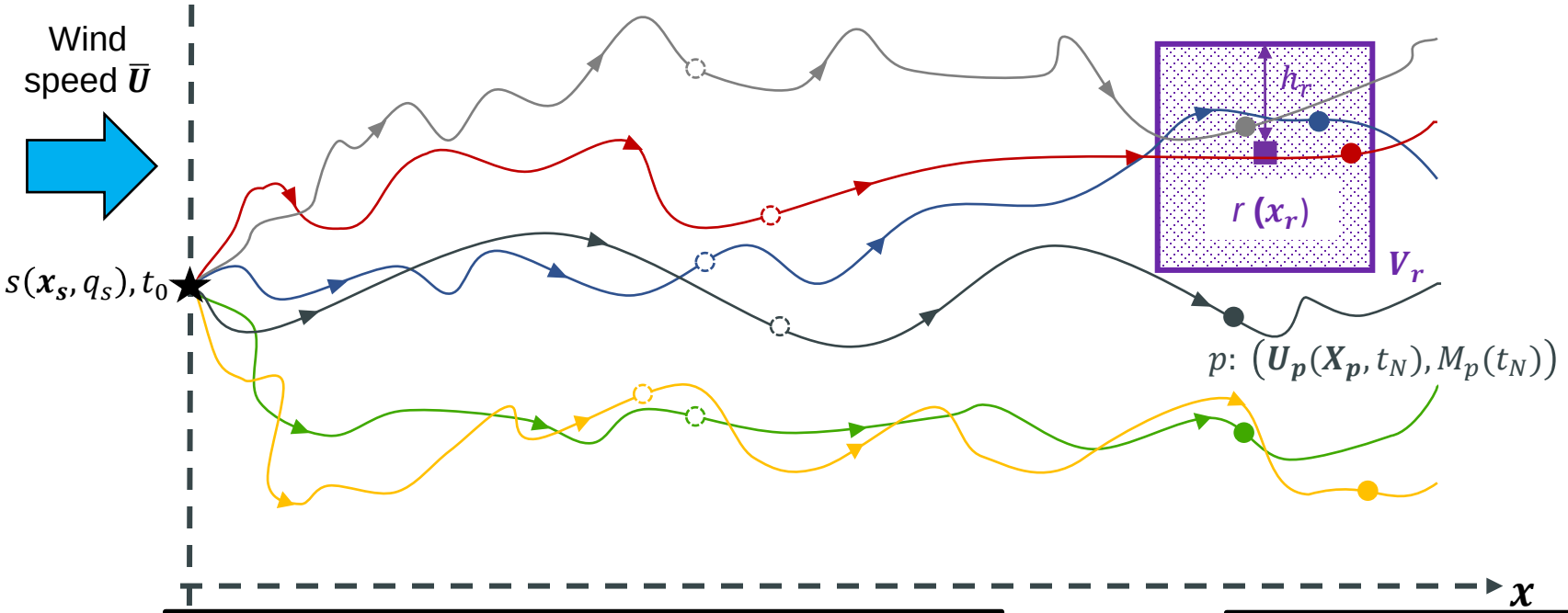
$$\underbrace{\nabla J}_{n_m\times1} = \underbrace{\left(\sum_{t=1}^{N}\sum_{p=1}^{N_p} O_{p,s}\right)}_{n_m\times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N\times1}$$

with $$\underbrace{O_{p,s} = P_{p,s}^T \left[\prod_{j=1}^{N-2} Q_{p,s}^{j+1\,T}\right] R_{p,s}^T}_{n_m\times n_N}$$

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

Adjoint computation-Aggregation step

Aggregation: Concentration computation at receiver r, after N iterations



$$\underbrace{\left(\frac{\partial f_s^N}{\partial u_s^{N-1}}\right)^T}_{n_{N-1}\times n_N} = \left[\underbrace{R_{1,s}}_{n_N\times 7} \cdots \underbrace{R_{p,s}}_{n_N\times 7} \cdots \underbrace{R_{N_p,s}}_{n_N\times 7}\right]^T$$

$R_{p,s}^T$ sensitivity matrix of concentration of particles at sensor r w.r.t. p characteristics at N^{th} iteration

$$\underbrace{\nabla J}_{n_m\times 1} = \underbrace{\left(\frac{\partial f_s^1}{\partial m_s}\right)^T}_{n_m\times n_1} \underbrace{\prod_{j=1}^{N-1}\left(\frac{\partial f_s^{j+1}}{\partial u_s^j}\right)^T}_{n_1\times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N\times 1}$$

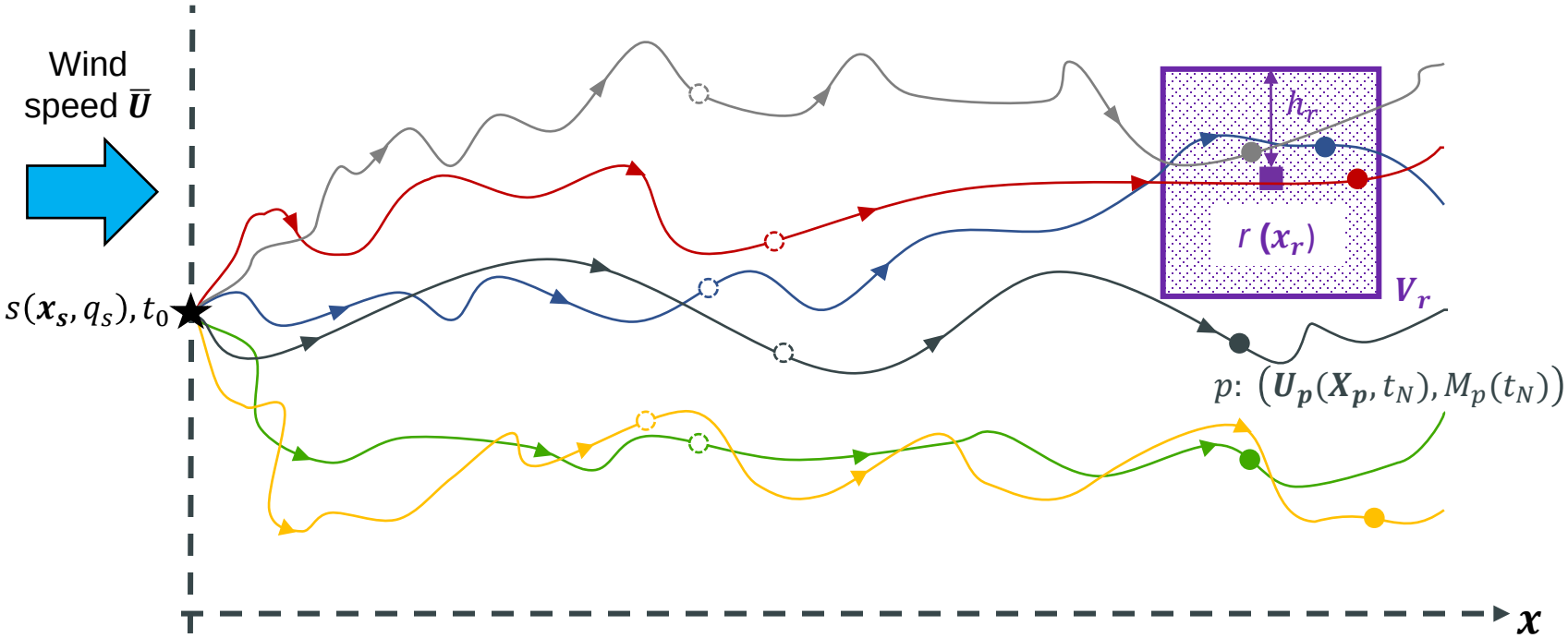
$$\underbrace{\nabla J}_{n_m\times 1} = \underbrace{\left(\sum_{t=1}^{N}\sum_{p=1}^{N_p} O_{p,s}\right)}_{n_m\times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}}\right]}_{n_N\times 1}$$

with

$$O_{p,s} = \underbrace{P_{p,s}^T\left[\prod_{j=1}^{N-2} Q_{p,s}^{j+1\,T}\right] R_{p,s}^T}_{n_m\times n_N}$$

APPLICATION TO A LAGRANGIAN STOCHASTIC PARTICLE DISPERSION MODEL

Adjoint computation



$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\sum_{t=1}^{N} \sum_{p=1}^{N_p} O_{p,s} \right)}_{n_m \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}} \right]}_{n_N \times 1}$$

- $O_{p,s}$ (size $4 * n_N$) computed independently for each particle

⇨ **Allows parallelization, reducing restitution time**

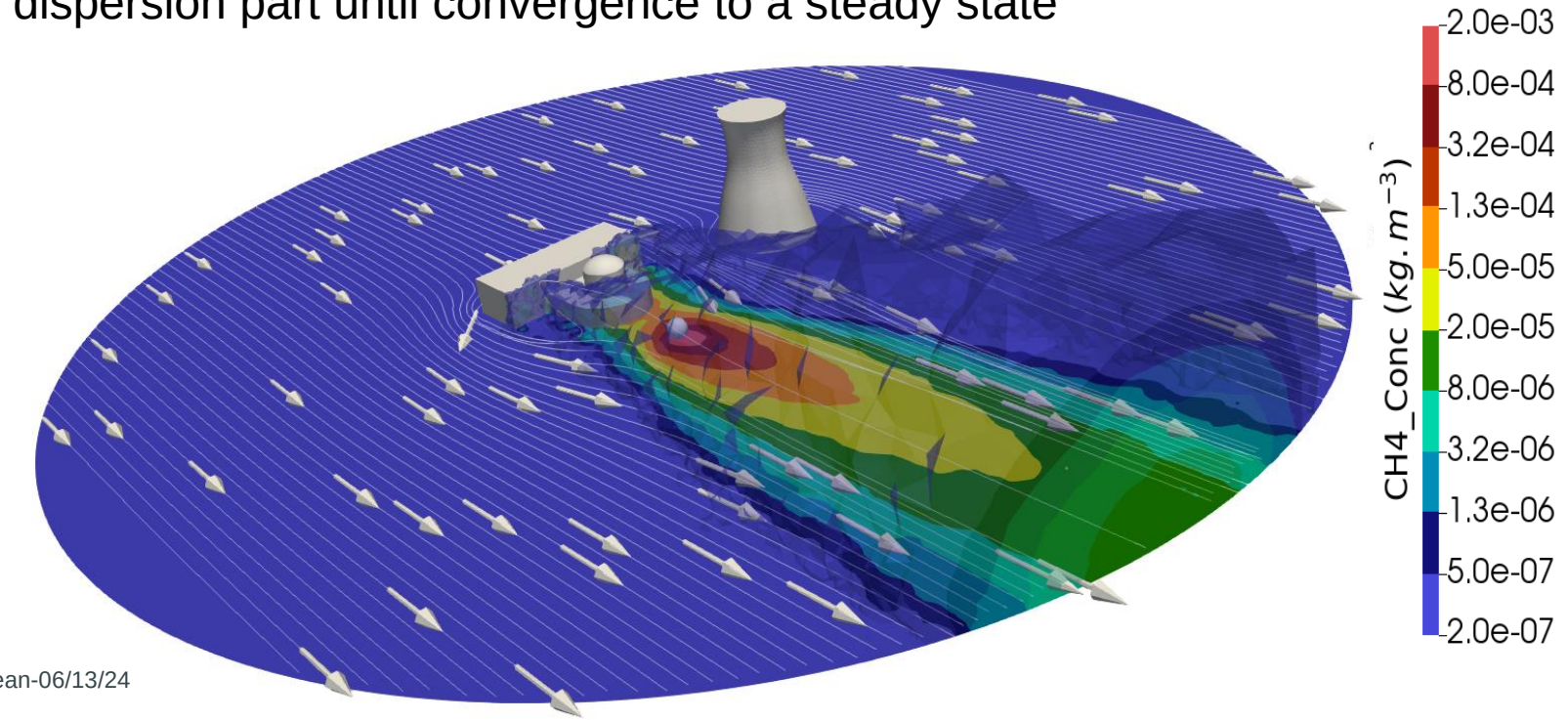
- Product calculated iteratively while forward model running

⇨ **Avoids particle data storage at each time step and excessive memory use**

- Easy adaptation of $O_{p,s}$ for more general cases, by summing equivalent terms corresponding to these cases.

APPLICATION RESULTS ON A NUMERICAL TEST CASE
Case description

- Computational domain: test case power plant at the centre of a 600-meter radius disk
- Continuous release from a point source located 10 meters above the ground
  - Steady configuration with Gaussian isotropic homogeneous turbulence and diagonal Reynolds stresses
  - Neutral atmosphere
- LS model runs dispersion part until convergence to a steady state



APPLICATION RESULTS ON A NUMERICAL TEST CASE
Adjoint concentration fields

- Direct concentration field and its adjoint ones to compute ground sensitivities
- **Gradients**: opposite directions of source displacement reducing difference between model and observations

⇨ **Allows to approach source true characteristics**

APPLICATION RESULTS ON A NUMERICAL TEST CASE
Adjoint concentration fields

- Direct concentration field and its adjoint ones to compute ground sensitivities
- **Gradients**: opposite directions of source displacement reducing difference between model and observations

⇨ **Allows to approach source true characteristics**

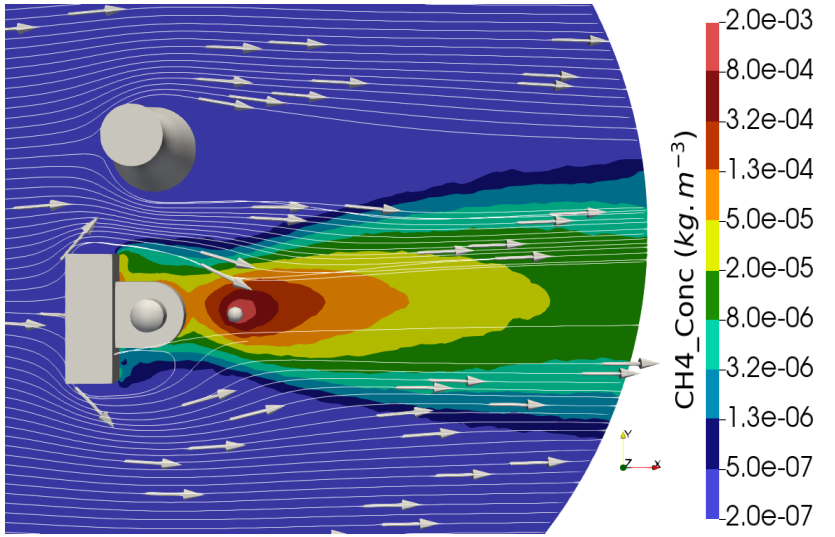


Figure 1 - Concentration field with wind flow (log-scale)

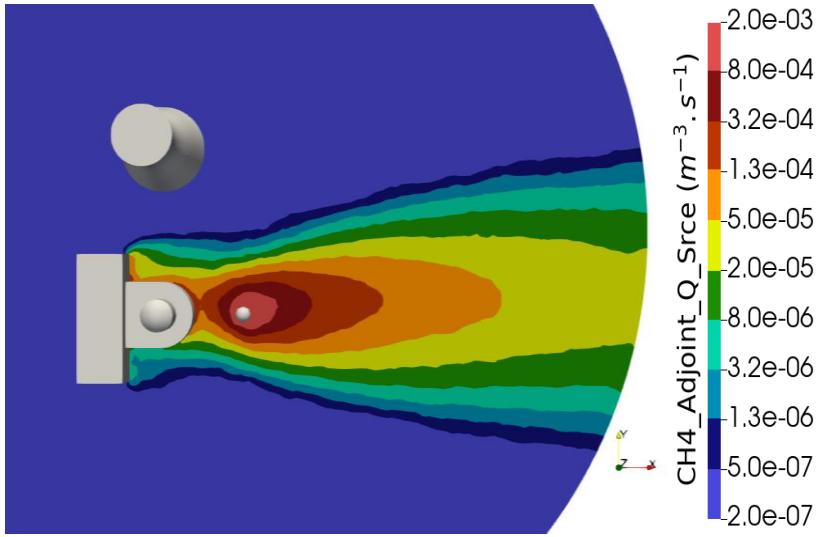


Figure 2 - Concentration variation for a unit increase of source flow rate (log-scale)

$$\underbrace{\nabla J}_{n_m \times 1} = \underbrace{\left(\sum_{t=1}^{N} \sum_{p=1}^{N_p} O_{p,s} \right)}_{n_m \times n_N} \underbrace{\left[\frac{u_s^N - d^N}{\overline{d^N} * \overline{d^N}} \right]}_{n_N \times 1} = 1$$

APPLICATION RESULTS ON A NUMERICAL TEST CASE
Adjoint concentration fields

- Direct concentration field and its adjoint ones to compute ground sensitivities
- **Gradients**: opposite directions of source displacement reducing difference between model and observations

⇨ **Allows to approach source true characteristics**

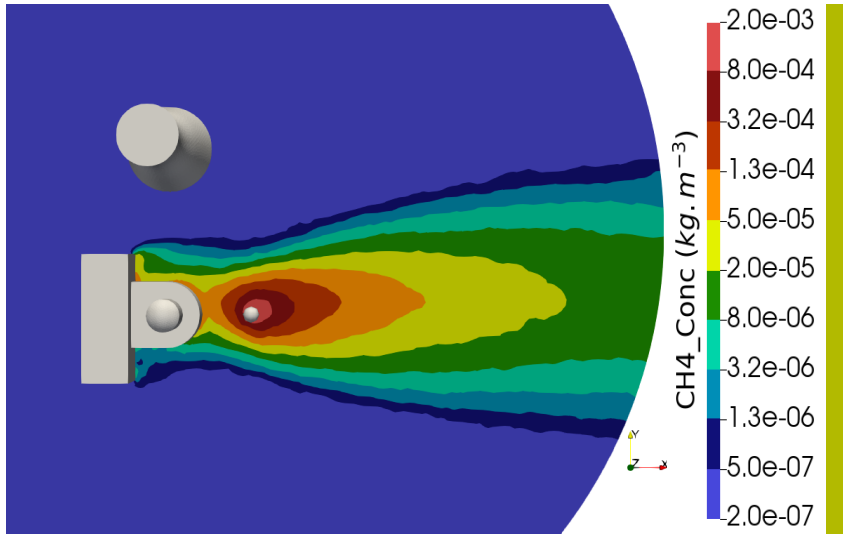


Figure 1 - Concentration field (log-scale)

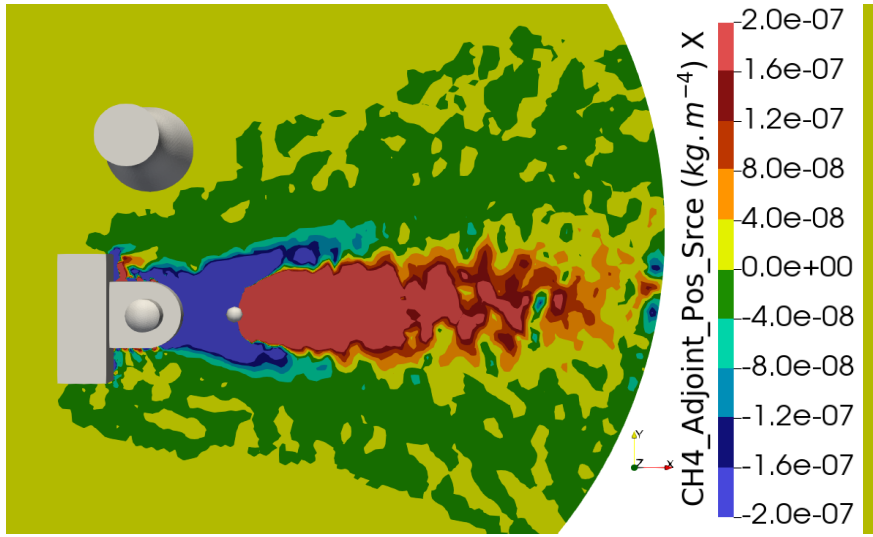


Figure 3 - Concentration variation for a unit increase of source X-coordinate (linear scale)

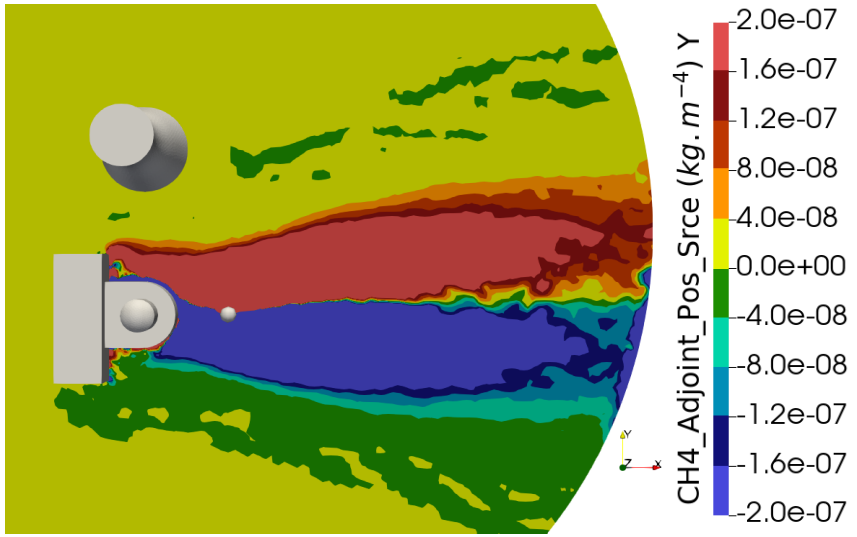


Figure 4 - Concentration variation for a unit increase of source Y-coordinate (linear scale)

CONCLUSION

- Use of the **adjoint state method** to solve the minimization problem avoids:
  - Computation of the whole Jacobian matrix
  - Dependence on the number of optimization parameters

⇨ **Save significant computation time !**

CONCLUSION

- Use of the **adjoint state method** to solve the minimization problem avoids:
  - Computation of the whole Jacobian matrix
  - Dependence on the number of optimization parameters

⇨ **Save significant computation time !**

- New application of the adjoint state method, for a LS model (Markovian explicit iterative model)

⇨ **Combined use of the adjoint method with a LS model is well suited to source characterization with real time constraint in complex industrial sites**

THANK YOU !