

Annexe — `projet-seirs`

31 janvier 2026

Table des matières

1	Vue d'ensemble	2
2	Arborescence du dépôt	2
2.1	Structure globale	2
2.2	Rôle des dossiers	2
3	Partie 1 — Modèle SEIRS par ODE (Python / C)	2
3.1	Codes sources	2
3.2	Commandes de génération	2
3.3	Fichiers de sortie (CSV) et origine	3
3.4	Figures (PNG) et origine	3
4	Partie 2 — Modèle Multi-Agent (Python / C++ / C)	3
4.1	Codes sources (simulation)	3
4.2	Scripts d'analyse (Python)	3
4.3	Commandes de génération (Partie 2)	4
4.4	Fichiers de sortie (CSV) et origine	5
4.5	Figures (PNG) et origine	5
5	Pipeline minimal de reproductibilité (récapitulatif)	5
5.1	Partie 1	5
5.2	Partie 2	6

1 Vue d'ensemble

Ce document décrit l'organisation du dépôt `projet-seirs` et précise :

- le rôle de chaque dossier,
- la description de chaque script (objectif, entrée/sortie),
- la liste des fichiers générés (CSV, figures) avec leur origine (script et commande).

2 Arborescence du dépôt

2.1 Structure globale

```
projet-seirs/
src/                      # Codes sources (Partie 1 et Partie 2)
data/                      # Sorties CSV (simulations et agrégations)
figures/                   # Figures PNG générées
notebooks/                 # Notebooks Jupyter d'analyse
rapport/                   # Rapport LaTeX/PDF (selon votre organisation)
```

2.2 Rôle des dossiers

Dossier	Role
src/	Contient l'ensemble des codes sources du projet : Partie 1 (ODE) et Partie 2 (SMA) en Python/C/C++.
data/	Contient les fichiers CSV générés par les simulations (répliques) et les agrégations (moyennes, pics).
figures/	Contient les figures PNG produites à partir des résultats (courbes, comparaisons, boxplots).
notebooks/	Contient les notebooks Jupyter utilisés pour analyser et comparer les sorties (Partie 1).
rapport/	Contient le rapport LATEX (et éventuellement les PDF) du projet.

3 Partie 1 — Modèle SEIRS par ODE (Python / C)

3.1 Codes sources

Fichier	Description
src/part1_seirs_ode/python/ seirs_part1.py	Implémente le modèle SEIRS continu et les méthodes numériques Euler explicite et RK4 en Python. Génère les CSV <code>python_euler.csv</code> et <code>python_rk4.csv</code> ainsi que les figures associées.
src/part1_seirs_ode/c/ seirs_part1.c	Implémentation équivalente en langage C (Euler explicite et RK4) produisant les CSV <code>c_euler.csv</code> et <code>c_rk4.csv</code> .

3.2 Commandes de génération

Python (Euler + RK4).

```
python3 src/part1_seirs_ode/python/seirs_part1.py
```

C (Euler + RK4).

```
gcc -O2 -Wall -Wextra -o seirs_part1_c src/part1_seirs_ode/c/seirs_part1.c -lm
./seirs_part1_c
```

3.3 Fichiers de sortie (CSV) et origine

Fichier généré	Généré par
data/part1_seirs_ode/ python_euler.csv	seirs_part1.py (méthode d'Euler en Python)
data/part1_seirs_ode/python_rk4.csv	seirs_part1.py (RK4 en Python)
data/part1_seirs_ode/c_euler.csv	seirs_part1.c (méthode d'Euler en C)
data/part1_seirs_ode/c_rk4.csv	seirs_part1.c (RK4 en C)

3.4 Figures (PNG) et origine

Figure générée	Générée par
figures/part1/python_euler.png	seirs_part1.py (tracé des courbes Euler à partir de python_euler.csv)
figures/part1/python_rk4.png	seirs_part1.py (tracé RK4 à partir de python_rk4.csv)
figures/part1/compare _lang_euler.png	notebooks/part1_analysis.ipynb (Python vs C, méthode Euler)
figures/part1/compare_lang_rk4.png	notebooks/part1_analysis.ipynb (Python vs C, méthode RK4)
figures/part1/compare_methods_ python.png	notebooks/part1_analysis.ipynb (Euler vs RK4 en Python)
figures/part1/compare_methods_ c.png	notebooks/part1_analysis.ipynb (Euler vs RK4 en C)

4 Partie 2 — Modèle Multi-Agent (Python / C++ / C)

4.1 Codes sources (simulation)

Fichier	Description
src/part2_multi_agent/python/ ma_seirs.py	Simulation SMA SEIRS en Python (réplication unique). Produit un CSV python_repXX.csv.
src/part2_multi_agent/cpp/ ma_seirs.cpp	Simulation SMA SEIRS en C++ (réplication unique) via binaire ma_seirs_cpp.
src/part2_multi_agent/c/ ma_seirs_c.c	Simulation SMA SEIRS en C (réplication unique) via binaire ma_seirs_c. Utilisée pour lancer 30 réplications (dossier c_runs/).

4.2 Scripts d'analyse (Python)

Script	Rôle
analyze_python_3reps.py	Calcule la moyenne des compartiments sur 3 réplications Python et génère CSV + figure.
analyze_cpp_3reps.py	Calcule la moyenne sur 3 réplications C++ et génère CSV + figure.
compare_py_cpp_means.py	Compare les moyennes Python vs C++ (courbe de $I(t)$) et génère une figure de superposition.
extract_peaks_part2.py	Extrait les indicateurs (jour du pic, valeur du pic) à partir des fichiers de réplications.
stats_part2_peaks.py	Calcule des statistiques descriptives et applique un test Kruskal-Wallis.
plot_part2_boxplots.py	Génère des boxplots pour peak_I et day_peak par langage.
plot_part2_illustrative_first100.py	Génère une figure illustrative <i>type sujet</i> : comparaison inter-langages de $I(t)/N$ sur 0–100 jours.

4.3 Commandes de génération (Partie 2)

C++ (compilation).

```
g++ -O2 -std=c++17 -Wall -Wextra -o ma_seirs_cpp src/part2_multi_agent/cpp/ma_seirs.cpp
```

C (compilation).

```
gcc -O2 -std=c11 -Wall -Wextra -o ma_seirs_c src/part2_multi_agent/c/ma_seirs_c.c -lm
```

Exécuter 3 réplications C++.

```
./ma_seirs_cpp --seed 111 --out data/part2_multi_agent/cpp_rep01.csv
./ma_seirs_cpp --seed 222 --out data/part2_multi_agent/cpp_rep02.csv
./ma_seirs_cpp --seed 333 --out data/part2_multi_agent/cpp_rep03.csv
```

Exécuter 30 réplications C (dossier c_runs).

```
mkdir -p data/part2_multi_agent/c_runs
rm -f data/part2_multi_agent/c_runs/*.csv

for i in $(seq 1 30); do
    ./ma_seirs_c $((1000+i)) data/part2_multi_agent/c_runs/c_rep$(printf "%02d" $i).csv
done
```

Analyse et figures (Partie 2).

```
python3 src/part2_multi_agent/python/analyze_python_3reps.py
python3 src/part2_multi_agent/python/analyze_cpp_3reps.py
python3 src/part2_multi_agent/python/compare_py_cpp_means.py
python3 src/part2_multi_agent/python/extract_peaks_part2.py
python3 src/part2_multi_agent/python/stats_part2_peaks.py
python3 src/part2_multi_agent/python/plot_part2_boxplots.py
python3 src/part2_multi_agent/python/plot_part2_illustrative_first100.py
```

4.4 Fichiers de sortie (CSV) et origine

Fichier généré	Généré par (origine)
data/part2_multi_agent/python_rep01.csv	ma_seirs.py (réplication Python #01)
data/part2_multi_agent/python_rep02.csv	ma_seirs.py (réplication Python #02)
data/part2_multi_agent/python_rep03.csv	ma_seirs.py (réplication Python #03)
data/part2_multi_agent/cpp_rep01.csv	ma_seirs_cpp (C++, seed=111)
data/part2_multi_agent/cpp_rep02.csv	ma_seirs_cpp (C++, seed=222)
data/part2_multi_agent/cpp_rep03.csv	ma_seirs_cpp (C++, seed=333)
data/part2_multi_agent/c_runs/c_rep01.csv ... c_rep30.csv	ma_seirs_c (C, 30 réplications; graines 1001..1030)
data/part2_multi_agent/python_mean_3reps.csv	analyze_python_3reps.py (moyenne sur 3 réplications Python)
data/part2_multi_agent/cpp_mean_3reps.csv	analyze_cpp_3reps.py (moyenne sur 3 réplications C++)
data/part2_multi_agent/python_peaks.csv	extract_peaks_part2.py (pics Python)
data/part2_multi_agent/cpp_peaks.csv	extract_peaks_part2.py (pics C++)
data/part2_multi_agent/c_peaks.csv	extract_peaks_part2.py (pics C, 30 réplications)

4.5 Figures (PNG) et origine

Figure générée	Générée par (origine)
figures/part2/python_mean_3reps.png	analyze_python_3reps.py (courbes S, E, I, R moyennes Python)
figures/part2/cpp_mean_3reps.png	analyze_cpp_3reps.py (courbes S, E, I, R moyennes C++)
figures/part2/compare_python_cpp_Imean.png	compare_py_cpp_means.py (superposition $I(t)$ moyen Python vs C++)
figures/part2/part2_boxplots_peaks.png	plot_part2_boxplots.py (boxplots des pics par langage)
figures/part2/compare_languages_I_first100days.png	plot_part2_illustrative_first100.py (figure illustrative $I(t)/N$ sur 0–100 jours)

5 Pipeline minimal de reproductibilité (récapitulatif)

Cette section résume une séquence de commandes permettant de régénérer les principaux résultats.

5.1 Partie 1

```
python3 src/part1_seirs_ode/python/seirs_part1.py

gcc -O2 -Wall -Wextra -o seirs_part1_c src/part1_seirs_ode/c/seirs_part1.c -lm
./seirs_part1_c
# Comparaisons: notebooks/part1_analysis.ipynb (ouvrir avec VS Code / Jupyter)
```

5.2 Partie 2

```
# C++
g++ -O2 -std=c++17 -Wall -Wextra -o ma_seirs_cpp src/part2_multi_agent/cpp/ma_seirs.cpp
./ma_seirs_cpp --seed 111 --out data/part2_multi_agent/cpp_rep01.csv
./ma_seirs_cpp --seed 222 --out data/part2_multi_agent/cpp_rep02.csv
./ma_seirs_cpp --seed 333 --out data/part2_multi_agent/cpp_rep03.csv

# C (30)
gcc -O2 -std=c11 -Wall -Wextra -o ma_seirs_c src/part2_multi_agent/c/ma_seirs_c.c -lm
mkdir -p data/part2_multi_agent/c_runs
for i in $(seq 1 30); do
    ./ma_seirs_c $((1000+i)) data/part2_multi_agent/c_runs/c_rep$(printf "%02d" $i).csv
done

# Analyses / figures
python3 src/part2_multi_agent/python/analyze_python_3reps.py
python3 src/part2_multi_agent/python/analyze_cpp_3reps.py
python3 src/part2_multi_agent/python/compare_py_cpp_means.py
python3 src/part2_multi_agent/python/extract_peaks_part2.py
python3 src/part2_multi_agent/python/stats_part2_peaks.py
python3 src/part2_multi_agent/python/plot_part2_boxplots.py
python3 src/part2_multi_agent/python/plot_part2_illustrative_first100.py
```