

# **Modélisation de propagation d'un virus**

*Martin Lehoux et Paul Cacheux  
À CentraleSupélec*

# Objectifs

- Comprendre les phénomènes régissant l'évolution d'une maladie dans une population
- Enrichir la modélisation de la propagation virale
- Effectuer des simulations numériques pour mesurer l'impact des décisions humaines

# Hypothèses générales

- L'âge n'a pas d'influence
- Homogénéité permanente
- Les phénomènes suivent des lois géométriques
- Tous les enfant sont sains

# Plan de la présentation

I. Modèle SIR

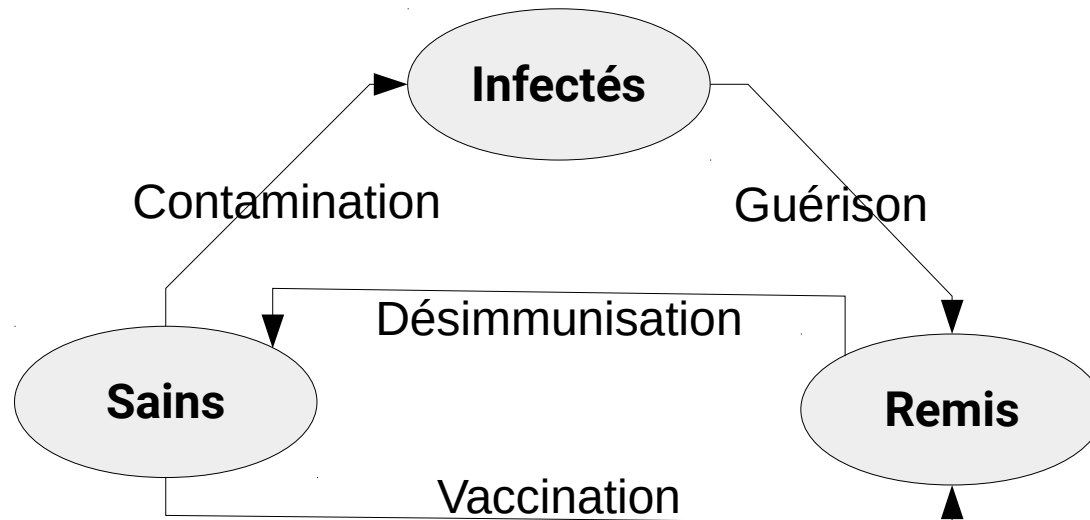
II. Modèle SEIQR

III. Limites

IV. Le code

# Modèle SIR

- Modéliser l'évolution d'une population {Sains, Infectés, Remis}



# Equations

$$s_{k+1} = \left(1 - death_{nat} - p \cdot \frac{i_k}{s_k + i_k + r_k}\right) \cdot s_k + loss \cdot r_k + birth \cdot (s_k + i_k + r_k)$$

$$i_{k+1} = p \cdot \frac{i_k}{s_k + i_k + r_k} \cdot s_k + (1 - death_{ill} - R) \cdot i_k$$

$$r_{k+1} = vacc_{rate} \cdot s_k + R \cdot i_k + (1 - death_{nat} - loss) \cdot r_k$$

# Equations

► Taux de mortalité (malade ou non) et de natalité

Population exposée  $s_{k+1} = (1 - \boxed{death_{nat}} - \boxed{p} \cdot \frac{i_k}{s_k + i_k + r_k}) \cdot s_k + \boxed{loss} \cdot r_k + \boxed{birth} \cdot (s_k + i_k + r_k)$

Population infectée  $i_{k+1} = \boxed{p} \cdot \frac{i_k}{s_k + i_k + r_k} \cdot s_k + (1 - \boxed{death_{ill}} - \boxed{R}) \cdot i_k$  ► Taux de guérison et de perte d'immunité

Population remise  $r_{k+1} = vacc_{rate} \cdot s_k + \boxed{R} \cdot i_k + (1 - \boxed{death_{nat}} - \boxed{loss}) \cdot r_k$

# Données numériques

## Exemple de valeurs des paramètres pour la smallpox

Paramètre	Valeur	Valeur normalisée (par jour)
<i>birth</i>	1,85 %/an	$5,0 \cdot 10^{-5}$
<i>death_nat</i>	0,78 %/an	$2,1 \cdot 10^{-5}$
<i>death_ill</i>	30 % en 24 jours	$1,3 \cdot 10^{-2}$
<i>R</i>	70 % en 24 jours	$2,9 \cdot 10^{-2}$
<i>loss</i>	Perte en 10 ans	$2,7 \cdot 10^{-4}$
<i>vacc</i>	80 % en 10 ans	$2,2 \cdot 10^{-4}$

### Transmission of smallpox,

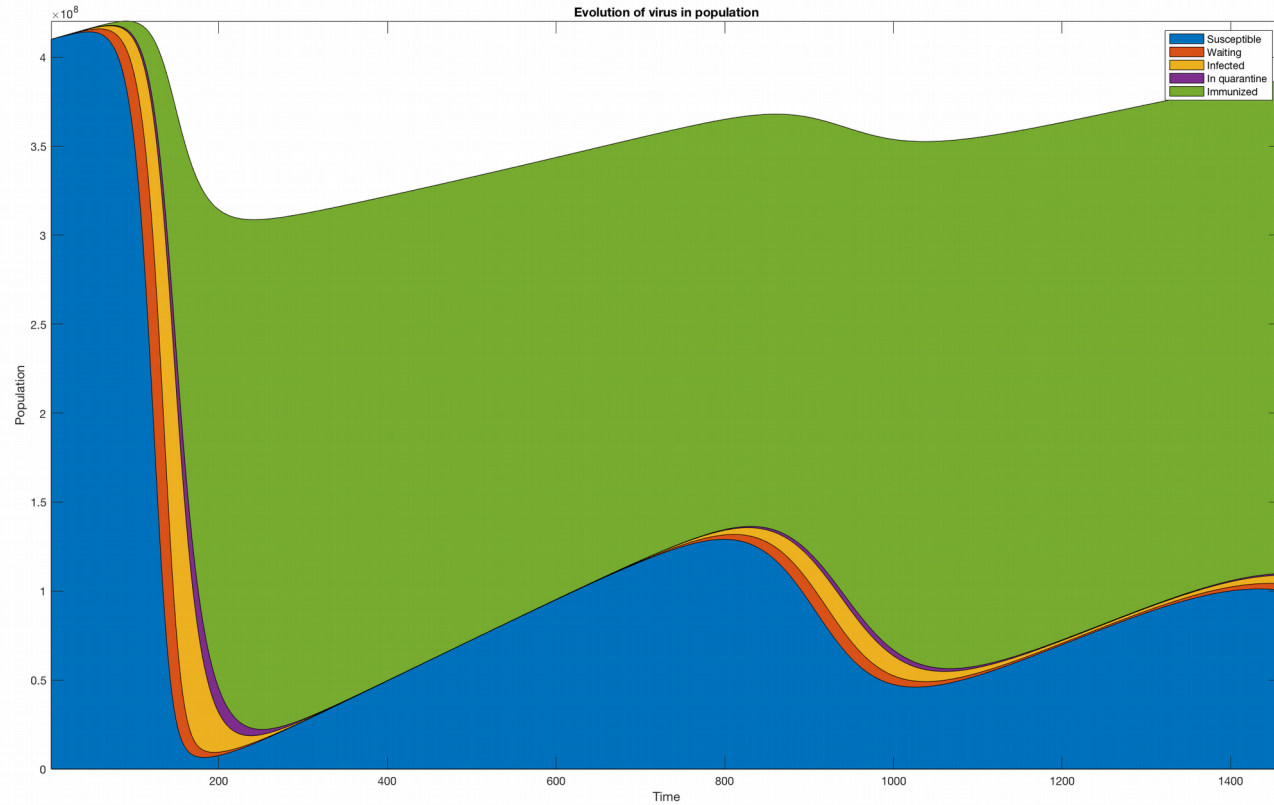
Center for disease control and prevention,

<https://www.cdc.gov/smallpox/clinicians/transmission.html>,

Décembre 2016



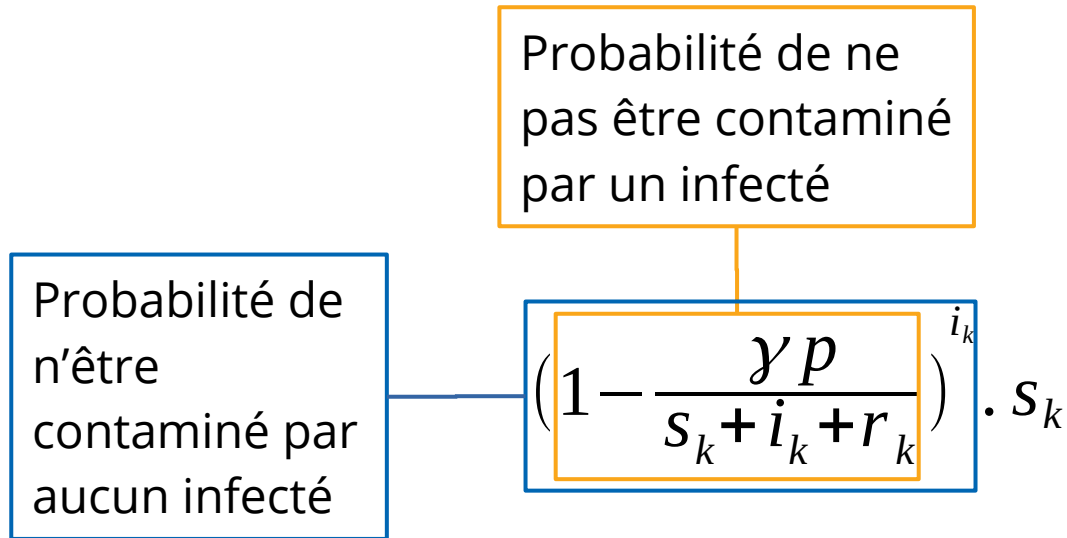
# Premiers résultats



# Problème rencontré

- ♦ **Problème** : le paramètre  $p$  ne se trouve pas simplement à partir des données
- ♦ Elaboration d'un nouveau modèle de contamination

# Modélisation de la contamination



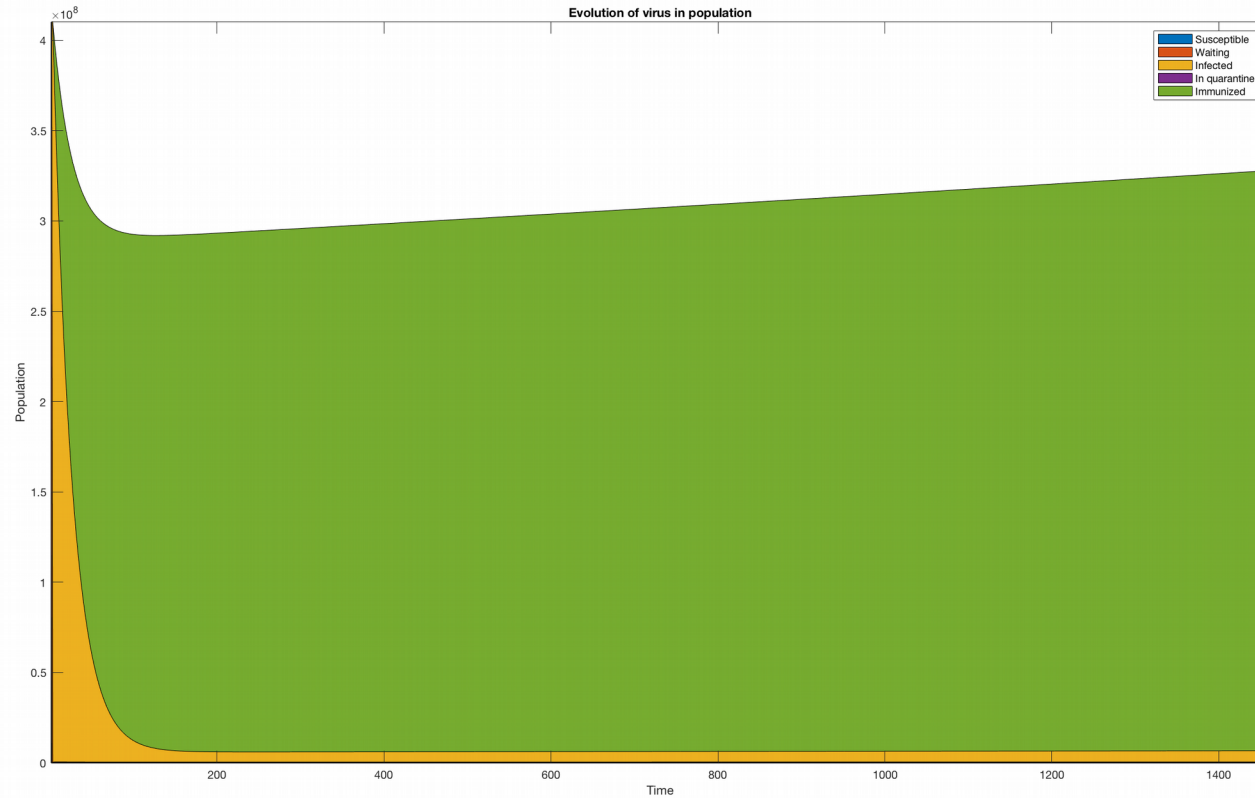
$\gamma = 25$  (*nombre de contacts par personne et par jour*)  
 $p$ : *proportion de contacts infectieux*

**Influence of social contact patterns and demographic factors on influenza simulation results**

Ruprecht Schmidt-Ott, Markus Schwehm and Martin Eichner,

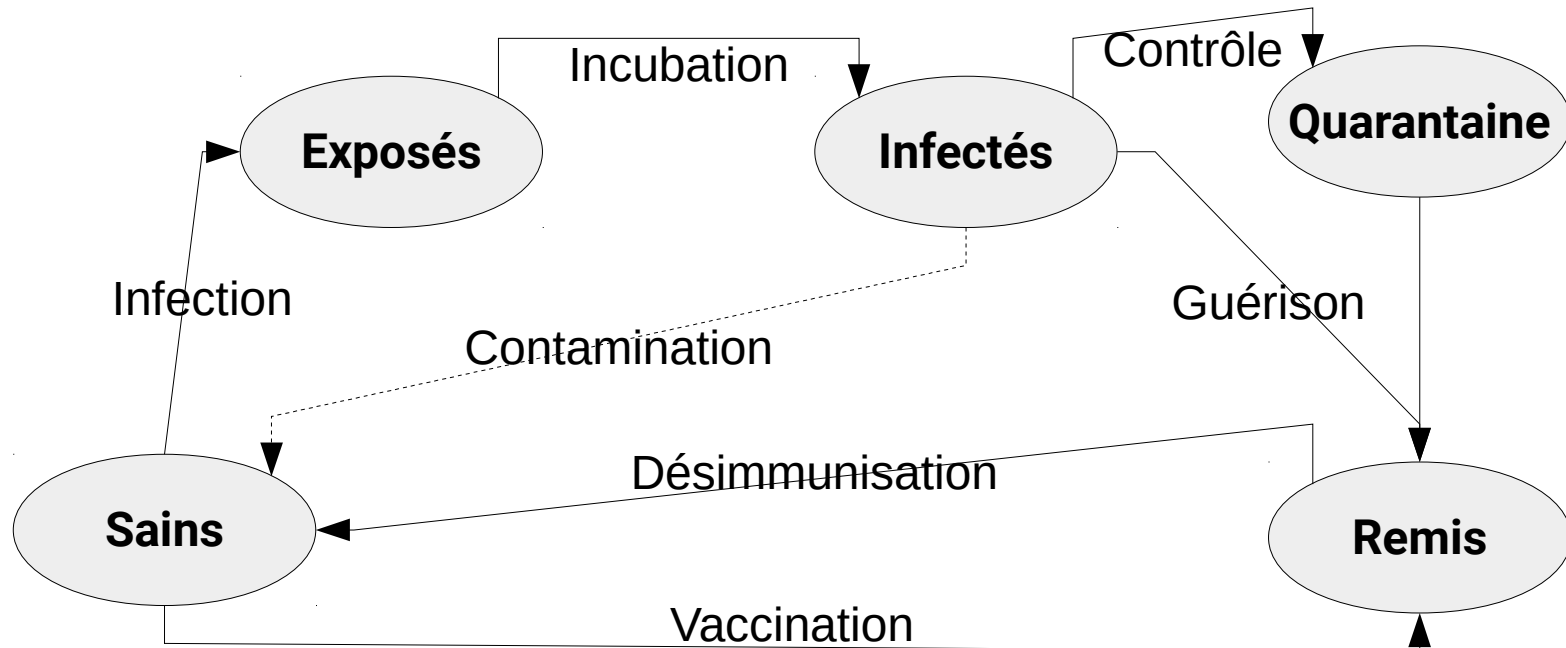
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5100331>, Novembre 2016

# Seconds résultats



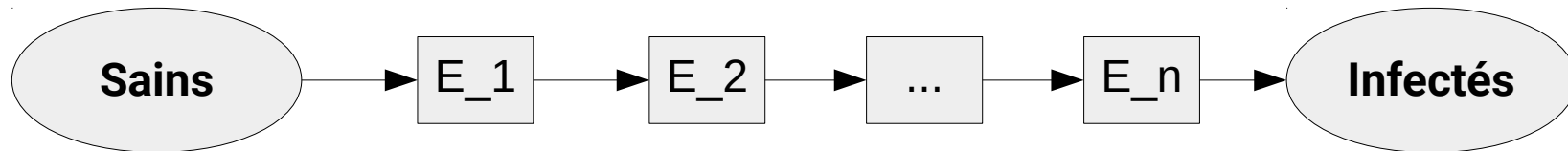
# Modèle SEIQR

- {Sains, Exposés, Infectés, Quarantaine, Remis}



# Modèle du retard pur

- Un compartiment par pas de retard
- Transfert total d'un compartiment à l'autre



**Application of Robustified Model Predictive Control to a Production-Inventory System,**  
C. Stoïca, M.R. Arahal, D.E. Rivera, P. Rodriguez-Ayerbe and D. Dumur

# Equations

$$s_{k+1} = \left(1 - death_{nat} - p \cdot \frac{i_k}{s_k + i_k + r_k}\right) \cdot s_k + loss \cdot r_k + birth \cdot (s_k + i_k + r_k)$$

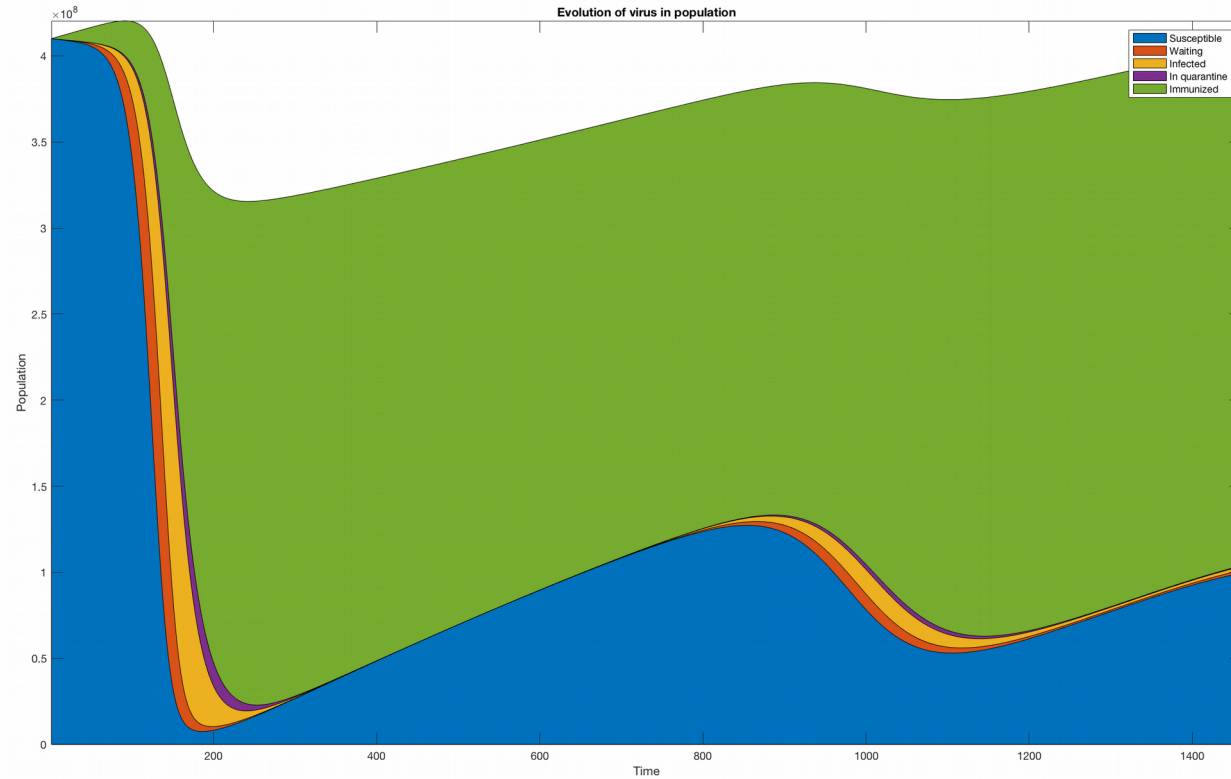
$$e_{1,k+1} = p \cdot \frac{i_k}{s_k + \sum e_{j,k} + i_k + r_k} \cdot s_k \quad e_{j,k+1} = e_{j-1,k}$$

$$i_{k+1} = e_{n,k} + (1 - death_{ill} - R - Q) \cdot i_k$$

$$q_{k+1} = Q \cdot i_k + (1 - death_{ill} - R) \cdot i_k$$

$$r_{k+1} = vacc_{rate} \cdot s_k + R \cdot i_k + (1 - death_{nat} - loss) \cdot r_k$$

# Nouveaux résultats





# Comportement

- ✗ Pas de changement de comportement lors de la maladie (nombre de contacts...)
- ✓ Paramètres différents pour chaque catégorie de population

# La maladie

- ✗ Profil de guérison constant
- ✗
- ✓ Diviser chaque compartiment en en créant un nouveau par pas (mais beaucoup de calculs à effectuer)

# Interactions

- ✕ Simulez des effets migratoires entre les pays
- ✓ Définition d'une fonction « score » et application des flux après chaque pas

# Le code

Trouvable sur :

[https://gitlab.viarezo.fr/kagamino/dynamique-populations/tree/master/SRI\\_matlab](https://gitlab.viarezo.fr/kagamino/dynamique-populations/tree/master/SRI_matlab)

Ou sur :

<https://bit.ly/2Me2qsd>

# Configuration - options.m

- Réglage des paramètres « fixes » de la simulation
- Classe avec une « property » par paramètre
- Est passée en argument à chaque « step »
- Matlab gui aisé grâce à cette extraction

# Le modèle – model.m

- Contient l'état à l'instant  $t$  du « réservoir » considéré et l'historique
- Une méthode de construction (avec l'état initial)
- Une méthode de « step »
- Simule un réservoir et un seul. Un tableau de « model » permet donc de simuler un ensemble de « réservoirs »

# Fonction de pas – step.m

- Permet d'extraire la fonctionnalité de pas
- Prend en paramètre l'état à  $t$  et les coefficients et retourne l'état à  $t+1$

# Evolution possible

- Simulation à plusieurs réservoirs : on « step » chacun des « model » puis on réalise les échanges entre réservoirs.
- Donc un « step » en  $O(n^2w)$
- Sans cette évolution le step est en  $O(w)$
- Avec  $n$  le nombre de réservoirs et  $w$  le nombre de jours d'incubation



# Conclusion

- Beaucoup de microphénomènes à modéliser pour atteindre des résultats proches de la réalité
- La modélisation numérique apporte des effets de bords

# Perspectives

- Tester différents scénarios de vaccination afin de voir l'ampleur de la réaction nécessaire pour détruire l'infection
- Tester différentes politiques de migration