CentraleSupélec - Première Année

Équations aux Dérivées Partielles - Projet FEniCS

Jeffrey KAIKATI (binôme avec Karim EL ASMAR)

Indication pour réussir à afficher les images du Notebook

Glissez les photos qui ont été transmises avec ce devoir sur cette page. Si cela ne suffit pas, assurez-vous que les photos sont dans le même dossier que le notebook. Vous pouvez me contacter par mail en cas de besoin : jeffrey.kaikati@student-cs.fr

Définition du Problème

La place du numérique croit tous les jours avec l'apparition de l'**Internet of Things** et du **Machine Learning**. Beaucoup d'informations sont alors partagées sur des bases de données et même sur des ordinateurs privés. Cependant, de nombreux scandales de data breach ont eu lieu ces dernières années à cause d'attaques par des logiciels maliciels plus connus sous le nom de **malwares**. Il est alors judicieux de s'intéresser à la propagation d'un tel logiciel.

Nous considérerons dans cette étude que l'ensemble des ordinateurs dans le monde peut être divisé en clusters, c'est-à-dire en groupes indépendants généralement facilitant le partage d'informations. Nous noterons alors:

- S_c les clusters **susceptibles d'être infectés** par le malware;
- M la **densité surfacique (par λ-u²) d'ordinateurs infectés par le *malware*** au jour j;
- ϕ le **flux surfacique (par λ-u²) du logiciel *malware***.

Avant toute considération physique pour décrire le phénomène de propagation du malware, nous devons préciser la métrique de notre espace. La distance euclidienne n'est pas adéquate dans notre problème : la proximité informatique de deux instances est plutôt liée à leur séparation par un nombre réduit de nœuds intermédiaires. La distance la plus naturelle à prendre en compte est alors la *distance naud*s définie par le nombre de nœuds strictement intermédiaires du plus court chemin entre les deux instances considérées. Nous admettrons pour la suite l'extension de la modélisation ci-dessous à notre cas. Il suffirait de faire des considérations continu-discret typiques des résolutions numériques.

Modélisation

Par analogie avec la loi de Fourier, nous pouvons expliciter la diffusion ayant lieu dans ce phénomène. Cependant, le coefficient de diffusion ne peut pas être considéré homogène et isotrope du fait des pertes à longues distances et d'autres facteurs que nous ne prendront pas en compte pour simplifier la résolution. Le coefficient de propagation λ sera alors défini par : $\lambda(x) = \Lambda(1 - ax + \frac{a(ax)^2}{2!})$ avec Λ réel positif décrivant une "puissance d'attaque" intrinsèque au *malware* considéré, a le coefficient de perte d'efficacité linéique et x la distance naud entre l'ordinateur et la source du *malware*. Ainsi, nous pouvons écrire l'équation de diffusion : $\phi(t,x) = -k(x)\nabla M(t,x) = \Lambda(1 - ax + \frac{a^2x^2}{2})\nabla M(t,x)$.

Ensuite, afin de parvenir à une équation aux dérivées partielles décrivant la situation, nous devons passer par une équation de conservation locale traduisant la variation du nombre d'ordinateurs infectés dans le temps à une certaine distance. L'équation s'écrit alors : $\frac{\partial M}{\partial t} + \nabla \cdot \phi = f$.

À présent, nous devons caractériser la fonction f. Celle-ci est plus compliquée à déterminer et la démarche statistique utilisée pour la trouver n'étant pas dans l'esprit de ce projet, nous allons admettre son écriture sous la forme $f(M,t) = n_i(t)M(E(x) - \frac{M}{K}) = n_i(t)ME(x) - n_i(t)M\frac{M}{K}$ où :

- $n_i(t)$ est la solution du problème de Cauchy :

$$\begin{cases} \frac{dn_i(t)}{dt} = -dn_i(t) + m \\ n_i(0) = m_0 \end{cases}$$

- K, d, m et m_0 des paramètres réels;
- $n_i(t)$ représente le facteur de décroissance du nombre de nœuds infectés;
- $E(x) = -(x - r)(x - s)$ représente à quel point la contagion du *malware* est modifiée en fonction de la distance naud*s*.

Cette forme de f semble cohérente : elle stipule que le taux de croissance des ordinateurs infectés dans un cluster est proportionnel à la densité actuelle des utilisateurs qui y sont infectés M. Il y a aussi compétition entre deux termes pour l'apparition/disparition du *malware*.

Équation aux dérivées partielles

À partir des expressions précédentes, nous obtenons l'équation aux dérivées partielles suivante :

$$\frac{\partial M}{\partial t}(t,x) = \frac{\partial(k\frac{\partial M}{\partial x})}{\partial x}(t,x) + (\frac{m}{d} - e^{-d}(\frac{m}{d} - m_0))M(t,x)E(x) - \frac{M(t,x)}{K}, \text{ avec } M^{'}(t,s) = \frac{\partial M}{\partial x}(t,s).$$

- Λ et a modélisent la diffusivité du *malware* ($\in (\mathbb{R}_+)^2$, le cas Λ=0 ou a=0 est inintéressant);
- r et s ($\in (\mathbb{R}_+)^2$) représentent respectivement l'évolution et le sommet de E;
- K ($\in \mathbb{R}_+^*$) représente la densité maximale se propageant sur le réseau;
- m ($\in \mathbb{R}_+$) représente le taux de nœuds infectés restant;
- m_0 ($\in \mathbb{R}_+$) représente le taux de nœuds initialement infectés;
- d ($\in \mathbb{R}_+^*$) représente le taux de décroissance du nombre de nœuds infectés.

Le but de ce projet est d'étudier une équation elliptique. Nous nous placerons donc dans **le cas stationnaire à 1D** pour lequel $\frac{\partial M}{\partial t} = 0$. L'éventuelle solution sera notée M_{st}.

Afin que notre problème soit bien posé, il est en partie nécessaire d'imposer des conditions aux limites.

Conditions aux limites du domaine

Comme nous l'avons indiqué en introduction, nous considérerons un cluster S_c pour x un nœud infecté. On s'intéressera dans la suite du projet à la résolution de l'équation *sur* $S_c =]l, L[$ (La modélisation est supposée *continue*[2]).

Les flux entre clusters seront considérés nuls : en effet, il est peu probable qu'une attaque du réseau de la Banque de France affecte celui d'une école au Brésil.

Les **conditions aux bords de type Neumann** sont alors les suivants:

$$\begin{cases} \frac{\partial M}{\partial x}(l,t) = 0 \\ \frac{\partial M}{\partial x}(L,t) = 0 \end{cases}$$

De plus la source et son entourage sont les plus exposés au *malware*, on pourrait définir une **densité de contamination à la source** : $M_{st}(l)$ (condition au bord de type **Dirichlet**).

Modélisation aléatoire

Le *malware* va rencontrer différents types de cibles. Celles-ci seront plus ou moins difficiles à infecter. Le réseau de la Banque de France, par exemple, est clairement plus sécurisé que celui d'une école, d'où l'importance de raisonner en terme de cluster et d'avoir des paramètres aléatoires.

Il est légitime de remplacer a, d, m, r, s et Λ par des **variables aléatoires** puisque ces paramètres sont difficiles à estimer.

On s'intéresse à l'équation adimensionnée : $-\frac{k}{L^2}\frac{\partial(k\frac{\partial M^*}{\partial x})}{\partial x} = \frac{m}{d}M_{st}(E - \frac{M_{st}}{K})$ sur $]0,1[$.

On choisit les modèles suivants inspirés de la documentation bibliographique :

- d suivant une loi $\mathcal{N}(10; 5)$;
- m suivant une loi $\tilde{\mathcal{N}}(0.05, 10^{-3})$;
- a suivant une loi uniforme sur $]0, 1[$;
- r suivant une loi uniforme sur $]0, 1[$;
- s suivant une loi uniforme sur $]0, 2[$;
- Λ suivant une loi uniforme sur $]0, 5[$;
- $K = 45$;
- $L = 120$;
- $M_{st}(0) = 30$.

NB : K est fixé par les contraintes du cluster considéré.

Simulation Fenics

In [1]:
```
!conda config --add channels conda-forge
!conda install -y -v fenics
```

```
Warning: 'conda-forge' already in 'channels' list, moving to the top
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done
Initializing UnlinkLinkTransaction with
  target_prefix: /Users/jeffreykaikati/anaconda3
  unlink_precs:

  link_precs:



# All requested packages already installed.
```

In [2]:
```
# Importation des bibliothèques

from fenics import *
from math import *
import matplotlib.pyplot as plt
import numpy as np
```

In [3]:
```
# Constantes de l'Equation (variables aléatoires)
alpha = 1*np.random.rand()
d = (np.random.randn(1)*5+100)[0]
r = 1*np.random.rand()
s = 2*np.random.rand()
D = 5*np.random.rand()  # Grand lambda
m = (np.random.randn(1)*0.001+0.05)[0]
K = 45
L = 120
M_0 = 30
```

In [4]:
```
# Definition de l'espace 8h et du maillage
maillage = UnitIntervalMesh(50)
H_P1 = FunctionSpace(maillage, 'P', 1)
```

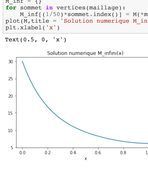
In [5]:
```
# Forme Variationnelle de l'Equation

Phi = TestFunction(H_P1)
M = Function(H_P1)


# Définition de fonction auxillaire
class MyExpressionJ(UserExpression): # le flux est note j sur le code
    def eval(self, value, x):
        dx = x[0]
        value[0] = D*(1-alpha*dx+(alpha*dx)**2/2)/(L**2) # formule du flux adimensionné
delta = MyExpressionJ(element=H_P1.ufl_element()) # conversion en ufc-element pour compatibilité


class MyExpressionK(UserExpression):
    def eval(self, value, x):
        dx = x[0]
        value[0] = -(dx-s)*(dx-r) #E(X)
E = MyExpressionK(element=H_P1.ufl_element()) #conversion en ufc-element


F = delta*dot(grad(M), grad(Phi))*dx - (m/d)*M*(E-M/K)*Phi*dx #Equation
J = derivative(F,M)
```

In [6]:
```
# Utilisation du module fenics pour enfin résoudre l'équation
class Dirichlet_0(SubDomain):
    def inside(self, x, on_boundary):
        booleen = (on_boundary and near(x[0],0.)*10**(-14))
        return booleen


condition_lim = DirichletBC(H_P1, M_0, Dirichlet_0())

solve(F==0, M, condition_lim, J=J)
```

```
Calling FFC just-in-time (JIT) compiler, this may take some time.
Calling FFC just-in-time (JIT) compiler, this may take some time.
```

In [7]:
```
# Graphe de M_infini
M_inf = []
for sommet in vertices(maillage):
    M_inf[(1/50)*sommet.index()] = M(*maillage.coordinates()[sommet.index()])
plot(M,title = 'Solution numerique M_infini(x)')
plt.xlabel('x')
```

Out[7]:
```
Text(0.5, 0, 'x')
```



Conclusion et exploitation de la simulation

Après avoir executé un grand nombre de fois la simulation, nous pouvons affirmer le comportement majoritairement **décroissant** de M_{st} quelles que soient les valeurs des variables aléatoires. La contribution de ces paramètres fournit des variations plus ou moins brusques des effectifs infectés.

Ce comportement est attendu puisque dans une limite de très longue durée, les ordinateurs se trouvant très loin de la source du *malware* (au sens de la *distance naud*s) sont à l'abris de la menace. **Les clusters les plus éloignés sont alors ceux qui présentent à priori le moins de cas d'infection**. Néanmoins, cela n'est pas toujours le cas. En effet, nous pouvons observer des situations différentes comme dans la capture d'écran ci-dessous.

Nous pouvons remarquer une augmentation de l'effectif infecté loin de la source. Ceci témoigne d'**une compétition existant entre la décroissance des effectifs à cause de l'éloignement de la source primaire et leur croissance dûe à l'infection par des sources secondaires**. Cette compétition est alors un phénomène très intéressant puisqu'elle décrit de manière plus complète le phénomène. L'attaque par un *malware* est contrée par des logiciels anti-virus ou une activité de cyber-sécurité mais reste néanmoins très invasive et réussit à toucher un grand nombre de nœuds.

D'autres observations assez intéressantes peuvent être faites. En modifiant le terme source grâce à r et s, nous pouvons contrôler la **position du maximum de la parabole** définie par E. Ainsi, en comparant les résultats donnés pour $(r, s) = (0.01, 0.1)$ représenté sur l'image de **gauche** et $(r, s) = (0.401, 0.5)$ représenté sur l'image de **droite**, nous pouvons apprécier des vitesses de décroissance très différentes.

Heuristiquement, la propagation du malware est **renforcée** loin (respectivement proche) de la source primaire, ce qui rend l'effectif infecté plus (respectivement moins) élevé en un grand x. Évidemment, pour affiner cela, il a fallu faire un grand nombre de tests pour vérifier que cette conjecture était indépendante de la valeur des autres variables aléatoires.

Pour finir, j'aimerais souligner le fait que cette résolution a été faite dans le cas *stationnaire* qui apporte des résultats exploitables mais qui explique la propagation du malware au bout d'un temps assez long. Les problématiques de *data breach* font néanmoins partie de situations **urgentes** pour lesquelles une étude approfondie doit considérer des variations temporelles.

Sources bibliographiques

FEniCS

- Documentation :
  - https://fenicsproject.org/documentation/
  - https://fenicsproject.org/pub/tutorial/pdf/fenics-tutorial-vol1.pdf
- Getting started : http://www.users.math.umn.edu/~arnold/8445/fenics-getting-started.pdf
- **URL** : https://fenics.readthedocs.io/projects/ufl/en/latest/manual/form_language.html

Sujet

- Jia, Peng, Liu, Jiayong, Pang, Yong, Liu, Liang, and Liu, Luping. "Modeling and Analyzing Malware Propagation in Social Networks with Heterogeneous Infection Rates." Physica A: Statistical Mechanics and Its Applications 507 (2018): 240-54. Web.
- R. Lerman, and R. Ghosh, "Information Contagion: an Empirical Study of Spread of News on Digg and Twitter Social Networks," in Proceedings of International Conference on Weblogs and Social Media (ICWSM), 2010.
- Linhe Zhu, Hongyong Zhao, Xiaoming Wang, "Stability and bifurcation analysis in a delayed reaction-diffusion malware propagation model," ScienceDirect, 2017
- Wang X M, Ke Z B, Zhao X Q, et al. Reaction-diffusion modeling of malware propagation in mobile wireless sensor networks. Sci China Inf Sci, 2013, 56: 092303(18), doi: 10.1007/s11432-013-4977-4

(1) : La forme de $\lambda(x)$ provient d'un développement à l'ordre 2 d'une exponentielle décroissante servant à une autre modélisation. Attention, ceci n'est pas un développement limité puisque l'argument de l'exponentielle ne tend pas vers 0. Ce n'est qu'une modélisation différente *inspirée* du modèle exponentiel et qui pourrait y être comparé dans une étude plus approfondie.

(2) : Cette hypothèse est primordiale pour qu'on se place dans le contexte du cours à savoir un ouvert de $\mathbb{R}$.

In [ ]:
```
```