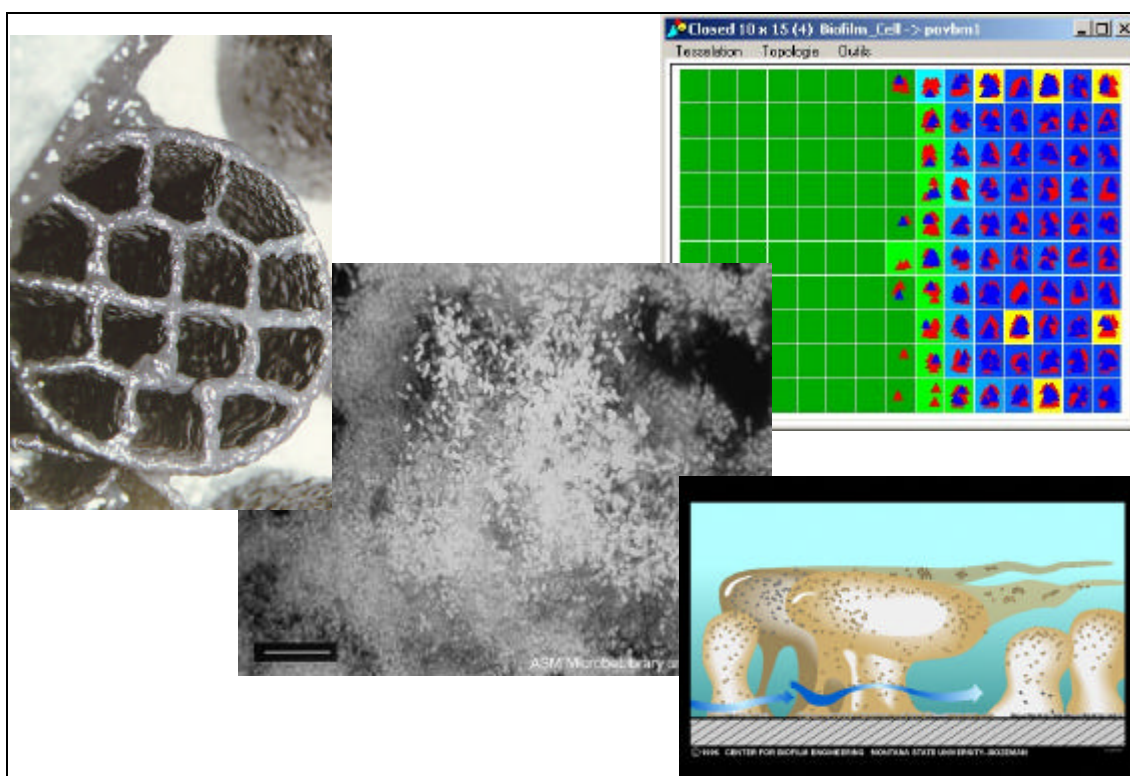


Modélisation des biofilms de digestion anaérobie par système multi-agents



Maîtres de stage :

Jean-Philippe Steyer (INRA LBE)
Christophe LePage (CIRAD TERA)

Responsable :

Francis Sevilla (Agro-M)

Laurent Lardon
Mémoire de DAA
AgroTIC
Octobre 2001

Remerciements

Je tiens tout d'abord à remercier tous ceux qui m'ont encadré et aidé au cours de ce travail, et particulièrement :

Les membres de l'équipe Green du CIRAD de Montpellier

Christophe LePage
François Bousquet
Pierre Bommel

Les membres de l'équipe automatique de l'INRA-LBE de Narbonne

Jean Philippe Steyer
Jérôme Harmand
Thierry Conte
Victor Alvaraz
Nancy et Pondichery
Tous les agents et stagiaires INRA

Sommaire

Remerciements	2
I. Introduction	5
II. Notations	7
III. Problématique	8
A. Présentation du problème	9
1. Structure d'accueil	9
2. La digestion anaérobie	9
3. La structure des biofilms	11
B. Les modèles déjà réalisés	13
1. Modèle global du réacteur	13
2. Modèles analytiques de biofilm	13
3. Automates cellulaires	14
IV. Elaboration du modèle	16
A. La modélisation multi-agents	17
1. Présentation	17
2. Plate formes de modélisation multi agent	17
B. Description du modèle SMA	19
1. Positionnement et but du modèle	19
2. Les différents phénomènes à modéliser	21
3. Interaction global / local	26
4. Architecture du modèle	27
5. Indicateurs développés	30
C. Adaptations préliminaires	30
1. Quel algorithme de diffusion - réaction	30
2. Désynchronisation des agents	33
3. Ajout d'un contrôleur	33
V. Validation – Evolution du modèle	34
A. Analyse des résultats	35
B. Etude de sensibilité	39
1. Conduite	39
2. Sensibilité des paramètres de modélisation	39
3. Sensibilité à la séquence des calculs	41
C. Développements ultérieurs du modèle	42
1. Développements du modèle	42
2. Limites de la plate-forme utilisée	44
VI. Conclusion	45
VII. Bibliographie	48
VIII. Annexes	49

Table des illustrations

<i>Figure 1 : Schéma de la digestion anaérobie</i>	10
<i>Figure 2 : Réacteur à lit fixe anaérobie</i>	11
<i>Figure 3 : Illustration de l'hétérogénéité spatiale d'un biofilm</i>	12
<i>Figure 4 : Modèle sur un digesteur parfaitement agité</i>	13
<i>Figure 5 : Bilan de matière et de biomasse d'un biofilm</i>	14
<i>Figure 6 : Architecture de la plate forme CORMAS</i>	18
<i>Figure 7 : Evolution des taux spécifiques acétogènes (μ_1) et méthanogènes (μ_2)</i>	23
<i>Figure 8 : Diagramme de décision pour la distribution de biomasse</i>	24
<i>Figure 9 : Illustration de la loi de Fick</i>	25
<i>Figure 10 : Diagramme de classes du modèle</i>	27
<i>Figure 11 : Diagramme d'exécution du modèle</i>	28
<i>Figure 12 : Diagramme d'exécution du modèle avec passerelle</i>	32
<i>Figure 13 : Profil de concentrations Simulation 1</i>	35
<i>Figure 14: Profils de concentrations Simulation 2</i>	36
<i>Figure 15 : Profils d'activité au sein du biofilm Simulation 1</i>	37
<i>Figure 16 : Profils d'activité au sein du biofilm Simulation 2</i>	37
<i>Figure 17 : Evolution des paramètres d'état vus du biofilm et vus du « réacteur » simulation 1</i>	38
<i>Figure 18 : Evolution des paramètres vus du biofilm et vus du réacteur simulation 2</i>	39
<i>Figure 19 : Règles de distribution de biomasse appliquées par Picioreanu</i>	43

I. Introduction

Qu'y a-t-il de commun entre une canalisation, une panse de ruminant et un réacteur de dépollution ? Le point commun qui fédère ce regroupement hétéroclite, ce sont les biofilms, ces amas de micro-organismes qui se forment sur toute surface exposée à l'eau. De fait ils sont omniprésents dans notre environnement : on en trouve sur les feuilles des algues, recouvrant les rochers des rivières, mais aussi sur les outils chirurgicaux où ils peuvent être pathogènes. A ces biofilms qu'on essaye tant bien que mal d'éliminer, il convient d'opposer les biofilms présents dans les réacteurs biologiques, qui réalisent efficacement les transformations chimiques industrielles ; les digesteurs utilisés pour la dépollution des eaux sont un exemple de l'utilisation fructueuse des biofilms.

Que ce soit pour les éliminer ou au contraire les maîtriser et les conserver, l'étude des biofilms s'avère aujourd'hui des plus utiles. De récentes techniques d'études ont permis de considérablement améliorer la connaissance de leur structure et des mécanismes en jeu dans leur formation et leur dynamique. La formation du biofilm revient à la formation d'un écosystème sommaire qui apporte une résistance aux perturbations de l'environnement (trophiques ou physiques) et permet une organisation en chaîne trophique.

Les digesteurs anaérobies utilisent la tendance des micro-organismes à former des biofilms. En maximisant la possibilité pour les bactéries de se fixer et d'effectuer efficacement les transformations biochimiques attendues, les réacteurs actuels permettent d'obtenir des taux de dépollution très intéressants. Mais en contrepartie de ces intérêts industriels et écologiques certains, ils souffrent de difficultés de contrôle et de problèmes de stabilité. La maîtrise des digesteurs passe sûrement par celle des écosystèmes constitués par le biofilm. C'est dans ce but que des modèles de biofilm sont réalisés.

Notre objectif a été d'utiliser une nouvelle approche de modélisation -la modélisation multi-agents- pour réaliser un modèle de biofilm capable de s'appuyer sur les mécanismes fondamentaux d'un biofilm plus que sur ses structures observées. Cette approche présente l'intérêt de mettre en évidence le manque de connaissances sur un point précis ou au contraire un niveau de connaissance suffisant pour reproduire les observations. Nous espérons aussi de cette approche qu'elle soit plus robuste qu'une approche classique et qu'elle permette de prévoir l'évolution du biofilm dans des conditions atypiques.

Ce rapport est organisé comme suit :

- la première partie présente la digestion anaérobie et les principales connaissances sur les biofilms, ainsi que les principales modélisations
- la deuxième partie présente la modélisation multi-agents, explique son intérêt pour la modélisation des biofilms, puis détaille l'élaboration du modèle, les différents mécanismes à modéliser ainsi que les solutions choisies
- la troisième partie présente l'analyse des résultats de simulation, l'analyse de sensibilité et les perspectives de développement du modèle.

II. Notations

Abréviations

AGV	Acides Gras Volatils
DOC	Demande chimique en Oxygène
TSH	Temps de séjour hydraulique
EPS	Extracellular Polysacharidic Substances
SMA	Système multi-agents

Lettres latines

X	Biomasse (g/L)
S	Concentration en substrat (g/L)
Ac	Concentration en acétate (g/L)
Q	Débit d'entrée
K _{s_i}	Constante de saturation (g/L)
k _{ij}	Coefficient de rendement de la réaction i pour la molécule j
A	Surface du biofilm
V _u	Volume utile du réacteur
J	Flux de matière
D	Coefficient de diffusivité

Lettres grecques

μ	Taux spécifique
γ	Taux d'arrachement
σ	Surface spécifique (m ² /m ³)

Indices

in	Entrée
out	Sortie
•	Définissant la solution du réacteur

III. Problématique

La digestion anaérobie est un procédé utilisé en dépollution des eaux usées. Ce procédé s'avère intéressant mais la maîtrise à l'échelle industrielle de réactions biologiques spontanées nécessite une bonne connaissance des phénomènes en jeu. Dans ce contexte, la modélisation de la digestion anaérobie permet d'améliorer la connaissance fondamentale de ce procédé mais aussi sa conduite industrielle et de synthétiser les connaissances actuelles. Cette partie présente le cadre du travail et ses objectifs. Après une présentation de la digestion anaérobie et des biofilms, les différents types d'approche et de modélisation sont brièvement décrits.

A. Présentation du problème	9
1. Structure d'accueil	9
2. La digestion anaérobie	9
3. La structure des biofilms	11
B. Les modèles déjà réalisés	13
1. Modèle global du réacteur	13
2. Modèles analytiques de biofilm	13
3. Automates cellulaires	14

A. Présentation du problème

1. STRUCTURE D'ACCUEIL

a) Le LBE

Le Laboratoire de Biotechnologie de l'Environnement (LBE) mène une recherche fondamentale et appliquée dans le domaine du traitement des pollutions et de la valorisation de la matière présente dans les rejets liquides et solides d'origine urbaine et agro-industrielle. Les travaux effectués portent sur la Microbiologie Moléculaire, le Génie Microbiologique, le Génie des Procédés et l'Automatique. Un centre de transfert rattaché au laboratoire permet de nombreuses relations avec les industries et les collectivités.

b) Approche Thématiques et projets de recherche de l'équipe automatique

L'Equipe Automatique, au sein de laquelle ce stage a été réalisé, a pour mission de

- Développer et mettre en œuvre des capteurs de mesure en ligne des principaux procédés et aider l'opérateur humain dans la gestion des installations,
- Utiliser les outils de l'Automatique pour améliorer la connaissance, la maîtrise et la conception des procédés biologiques de traitement des rejets issus de l'activité
-

Cette thématique fédère des aspects issus des Sciences pour l'Ingénieur et intègre les connaissances microbiologiques, technologiques et économiques dans une vision globale de

-
- ligne à finalité industrielle.
- Commande et optimisation de la conception des réacteurs : Développement et mise en œuvre de lois de commande pour gérer les incertitudes et les non-linéarités des procédés biologiques interconnectés.
- Aide à la décision : Développement et mise en œuvre de stratégies de détection et localisation de défauts et de diagnostic en temps réel.

A DIGESTION ANAEROBIE

La digestion anaérobie, appelée aussi méthanisation est un procédé spontané identifié depuis pour la dépollution des eaux usées seulement depuis les années 70

a) principe

Les principales étapes de la digestion anaérobie telles qu'elles ont été identifiées par Zeikus (1980) font appel à différentes populations de micro-organismes.

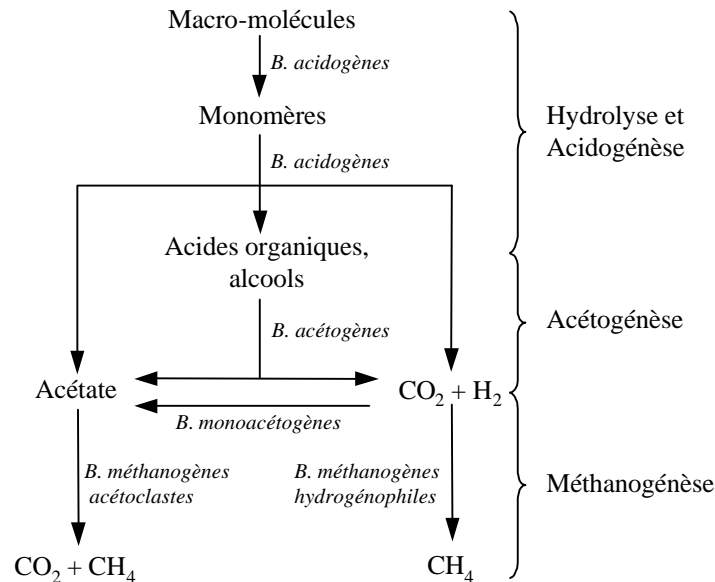


Figure 1 : Schéma de la digestion anaérobie

La première étape est assurée par des enzymes extracellulaires qui assurent la transformation du substrat en monomères (acides gras, glycérol, acides aminés, sucres simples). La deuxième phase acidogène métabolise les monomères en acides organiques à courtes chaînes. Ceux-ci sont alors métabolisés en **Acides Gras Volatils (AGV)**. L'étude thermodynamique des réactions acétogènes conduit à l'hypothèse que les populations acétogènes sont inhibées par leur produit. Leur croissance dépend donc de l'élimination de leurs produits finaux (*i.e.* Acides gras volatils et H_2) par d'autres micro-organismes.

Les micro-organismes méthanogènes sont des *archéobactéries*, leur vitesse de croissance est beaucoup plus faible que celle des acétogènes. Elles métabolisent les produits des bactéries acétogènes et acidogènes, produisant du méthane et du dioxyde de carbone. Elles sont toutefois inhibées par leur substrat.

Cette description volontairement simpliste des réactions complexes de la digestion anaérobie suffit à faire apparaître un point fondamental : le réacteur ne dépolluera efficacement que si les deux catégories de micro-organismes travaillent de concert mais l'obtention d'un équilibre entre les acétogènes et les méthanogènes n'est pas automatique ; on pourra noter par exemple que si la concentration en substrat est trop forte (cas de surcharge du réacteur), la forte production d'AGV qui s'en suit provoque l'inhibition des méthanogènes, puis celle des acétogènes, bloquant le réacteur.

b) Intérêts et limites de la digestion anaérobie

La digestion anaérobie offre plusieurs avantages par rapport à la voie plus classique aérobie : capacité à dégrader des effluents complexes et concentrés, faible production de boues, faibles coûts de fonctionnement voire production d'énergie.

Mais ce procédé se montre instable, sensible aux variations des conditions opératoires et parfois plusieurs mois sont nécessaires pour ramener le réacteur à un point de fonctionnement normal.

Ce défaut majeur de la digestion anaérobie explique son faible emploi industriel et nécessite pour être franchi une amélioration des outils de diagnostic et de pilotage qui passe entre autre par une meilleure connaissance des phénomènes biologiques en jeux.

c) les grands types de réacteurs

La digestion anaérobie est mise en œuvre dans des réacteurs (ou fermenteurs) étanches. Aux réacteurs agités, où la biomasse n'était pas fixée et devait donc être récupérée par décantation, on préfère maintenant des réacteurs à biomasse fixée. L'immobilisation de la biomasse permet d'augmenter les performances et la robustesse d'un réacteur. Il existe plusieurs modalités d'immobilisation de la biomasse :

- Le filtre ou lit fixe consiste en un garnissage immobile colonisé par les bactéries ; il présente toutefois une grande sensibilité au colmatage.
- Le réacteur à lit de boue utilise la tendance naturelle de la biomasse à former des granules ; ce type de réacteur s'avère très efficace mais les phénomènes de granulation sont peu maîtrisés. C'est malgré tout le modèle le plus répandu en Europe.
- Le réacteur à lit fluidisé utilise des particules solides fluidisées par l'alimentation du réacteur comme support de croissance des bactéries ; ces réacteurs ne présentent pas de colmatage et sont très efficaces. Cette technique prometteuse n'est malheureusement qu'utilisée avec parcimonie au stade industriel du fait de ses difficultés opératoires

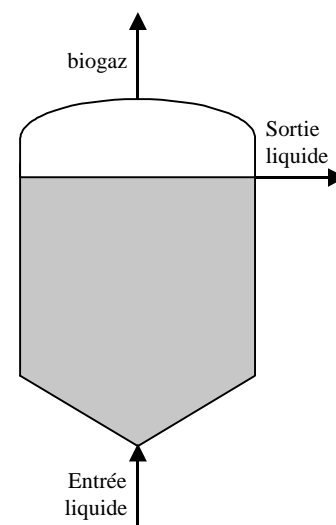


Figure 2 : Réacteur à lit fixe anaérobie

Le travail réalisé dans ce stage porte sur les lits fixes ; la disponibilité de nombreuses données et de modèles fiables sur un réacteur de ce type explique ce choix.

3. LA STRUCTURE DES BIOFILMS

Les micro-organismes s'organisent sur les supports et particules de son environnement en films adhérents. Le biofilm ainsi formé est le lieu privilégié des réactions biochimiques, on trouve en effet très peu de bactéries libres et actives dans la phase libre. Dans ce contexte,

mieux comprendre la mise en place et la dynamique des biofilms s'avère crucial pour améliorer le pilotage des digesteurs.

Un biofilm est constitué de bactéries fixées entre elles et au support par une matrice extracellulaire polysaccharidique (**Extracellular Polysaccharidic Substance**). Historiquement, les biofilms étaient d'abord considérés comme une structure homogène mais de nouvelles méthodes d'étude ont permis de montrer qu'ils sont en fait des milieux hétérogènes et stratifiés.

La surface du biofilm, révélée par microscopie électronique, présente des formes caractéristiques de champignon et de tulipe. De même au sein du biofilm, il existe de nombreuses lacunes et des canaux. Les mesures de flux par microscopie confocale laser et à l'aide de micro sondes, ont montré que ces canaux jouaient un rôle important dans la pénétration du biofilm par la solution en apportant un terme convectif.

Les microsondes ont par ailleurs mis en évidence la stratification des biofilms : au fur et à mesure que la profondeur augmente, la densité augmente alors que la porosité, la biomasse active et la diffusivité diminuent.

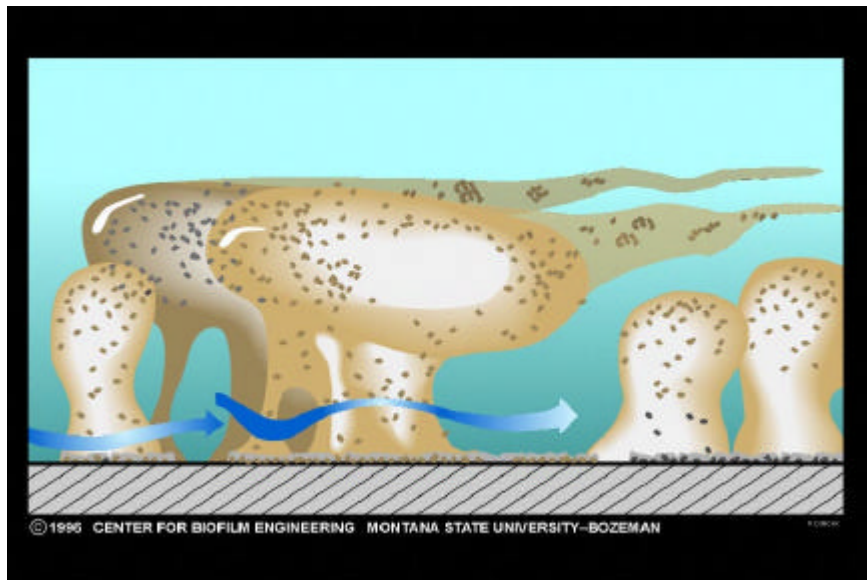


Figure 3 : Illustration de l'hétérogénéité spatiale d'un biofilm

Ces moyens d'investigation très précis des biofilms ont montré que les conditions du milieu jouent un rôle important sur leur structure. Les concentrations des solutés et les conditions hydrodynamiques, en particulier, en agissant sur l'équilibre entre la croissance et le détachement des cellules influent fortement la structure.

Des études sur les biofilms, on retiendra que de nombreux facteurs influent leur structure et leur développement :

- Des facteurs physico-chimiques : transfert des solutés, dégazage, contrainte hydrodynamique
- Des facteurs biologiques : expression des caractères génétiques pour l'adhérence cellulaire, la cinétique, la signalisation intercellulaire
- Des facteurs aléatoires : colonisation initiale, mutation

B. Les modèles déjà réalisés

Plusieurs approches sont possibles pour modéliser le réacteur. On peut adopter une approche globale, ou chercher à modéliser plus finement le biofilm.

1. MODELE GLOBAL DU REACTEUR

Le réacteur est considéré dans son ensemble, les variations des variables d'état du réacteur sont décrites en fonction des entrées et sorties, classiquement en s'appuyant sur des balances massiques des molécules et des biomasses. La dynamique du biofilm est peu ou pas explicitée, sa distribution spatiale en masse et en activité non décrite.

Ces modèles sont intéressants pour assister le contrôle du réacteur et prévoir l'évolution du réacteur dans son domaine normal d'utilisation. Les modèles de biofilm permettent de mieux décrire le fonctionnement intime du réacteur.

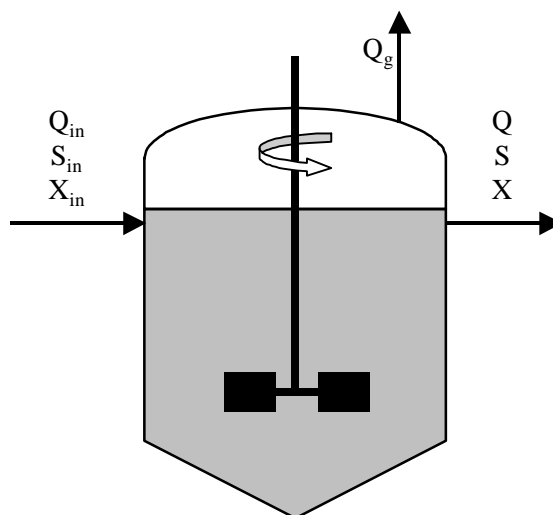


Figure 4 : Modèle sur un digesteur parfaitement agité

2. MODELES ANALYTIQUES DE BIOFILM

Les modèles analytiques consistent en la définition et résolution d'équations différentielles décrivant l'évolution d'un jeu de variables. Ils sont le plus souvent utilisés pour des systèmes de grande échelle où se dégage une certaine homogénéité. Jusqu'aux années 80, les biofilms étaient supposés être des structures homogènes planes, ce qui a conduit à la formulation de modèles mathématiques continus pour la prévision de l'activité des biofilms. La prise en compte de l'hétérogénéité des biofilms dans ce type de modèle a consisté à modéliser les processus d'attachement-détachement de cellules, et de convection par les pores.

Un modèle de biofilm classiquement décrit d'une part la dynamique des espèces chimiques, d'autre part celle des micro-organismes (cf. Figure 5).

Les molécules sont soumises à des phénomènes de convection, de diffusion, et de réaction. Ces phénomènes sont très rapides et atteignent leur point d'équilibre en moins d'une minute. Ils constituent la dynamique rapide du biofilm.

Les micro-organismes se multiplient, meurent ou sont inactivés, se détachent et s'attachent au biofilm. Ces phénomènes sont beaucoup plus lents et atteignent leur point d'équilibre au bout d'un temps beaucoup plus important (de plusieurs heures à la semaine). Ils constituent la dynamique lente du biofilm.

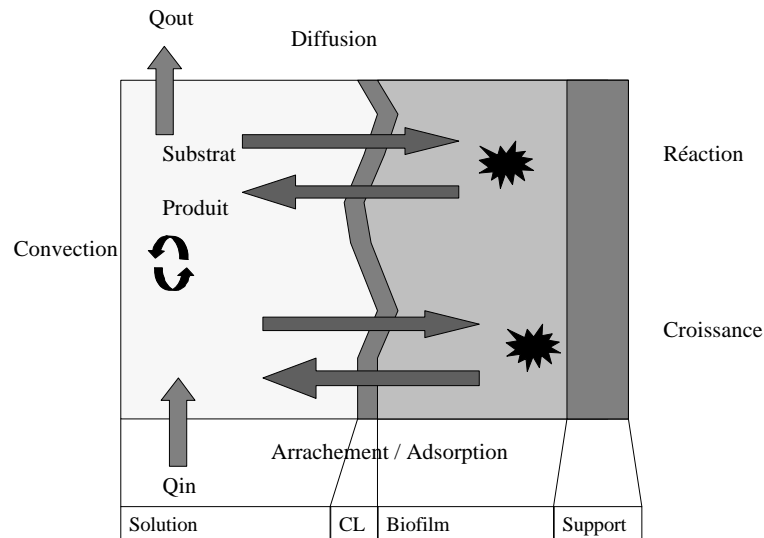


Figure 5 : Bilan de matière et de biomasse d'un biofilm

Les deux dynamiques du biofilm s'inscrivent dans des échelles de temps très différentes ; il est fréquent en modélisation des biofilms qu'on utilise cette propriété pour modéliser séparément les deux dynamiques : les calculs de chaque dynamique sont effectués en supposant fixes les variables affectées par l'autre dynamique [Picioreanu *et al* 97].

3. AUTOMATES CELLULAIRES

Les modèles discrets deviennent nécessaires dès que l'échelle est suffisamment petite pour que des phénomènes individuels deviennent importants.

Les automates cellulaires représentent un type de modèle discret, adapté à la modélisation de systèmes hétérogènes. Un automate cellulaire est un système où l'espace est divisé en cellules considérées comme des entités individuelles, qui peuvent prendre plusieurs états possibles. Elles changent d'état à des moments fixes selon une règle de transition fondée sur la configuration d'états au voisinage de chaque cellule. Il s'agit bien de modèles discrets puisque l'espace mais aussi le temps et les états possibles de chaque cellule sont discrets. Les règles de transitions sont généralement très simple mais aboutissent à des phénomènes macroscopiques complexes, comparables à ceux définis par les équations analytiques.

Ils ont été utilisés récemment pour la modélisation des biofilms [Picioreanu *et al* 1997].

Une autre approche de modélisation est le recours aux systèmes multi-agents. C'est la voie que nous avons choisi d'explorer. Cette approche encore très récente s'avère d'ores et déjà prometteuse car capable d'intégrer de nombreux phénomènes.

Conclusion :

La digestion anaérobie présente de nombreux intérêts économiques et environnementaux mais souffre de son instabilité. L'étude des biofilms vise à améliorer les connaissances sur le fonctionnement de ses réacteurs afin d'améliorer leur diagnostic et leur pilotage.

Notons que la modélisation des biofilms dépasse de loin le cadre seul des digesteurs. Les phénomènes mis en jeu dans un biofilm de digesteur sont semblables à ceux rencontrés dans une conduite d'eau ou dans le développement des endotheliums.

De nombreux modèles de biofilms ont déjà pu être formulés, soit à partir d'équations analytiques, soit à partir d'automates cellulaires. Nous avons choisi d'expérimenter une nouvelle voie de modélisation par système multi-agents qui devrait nous permettre d'améliorer l'utilisation des connaissances sur les biofilms pour comprendre et prévoir leur dynamique.

IV. Elaboration du modèle

Le modèle de biofilm réalisé s'appuie sur le formalisme multi-agent, apparu dans les années 80. Ce type de modélisation bien que très prometteur pour les simulations en écologie, reste encore peu utilisé. Après une brève introduction au formalisme multi-agents, les différents phénomènes mis en jeu dans le biofilm sont décrits ainsi que leur implémentation dans la plate forme CORMAS.

A. La modélisation multi-agents	17
1. Présentation	17
2. Plate formes de modélisation multi agent	17
B. Description du modèle SMA	19
1. Positionnement et but du modèle	19
2. Les différents phénomènes à modéliser	21
3. Interaction global / local	26
4. Architecture du modèle	27
5. Indicateurs développés	30
C. Adaptations préliminaires	30
1. Quel algorithme de diffusion - réaction	30
2. Désynchronisation des agents	33
3. Ajout d'un controleur	33

A. La modélisation multi-agents

1. PRESENTATION

Face à un système complexe, on réalise des modèles pour :

- faire apparaître des équilibres et des attracteurs entre les éléments,
- contrôler la dynamique voire la viabilité d'un système
- aider à la décision
- prédire les évolutions possibles.

Une première approche de modélisation consiste à décrire l'ensemble du système à l'aide d'un jeu d'équations. La modélisation multi-agent est une autre approche apparue dans les années 80 qui substitue à cette notion d'expert individuel celle d'action distribuée : on répartit les actions et les décisions en éléments simples. La modélisation multi-agent repose sur l'hypothèse que les propriétés d'un groupe d'individus sont déterminées par les comportements des individus qui le composent.

Un **Système Multi Agents (SMA)** est constitué typiquement de :

- *d'un ensemble d'Agents* : Un agent est une entité virtuelle dont le comportement est autonome, évoluant dans un environnement dont il a une perception partielle et sur lequel il peut agir et avec lequel il peut interagir (Demazeau 96). Il peut être purement réactif (il répond aux changements de son environnement) ou cognitif (mémorise et analyse ses perceptions et mobilise ses objectifs et croyances). Il est susceptible de communiquer et d'interagir avec son environnement.
- *d'un environnement* : L'environnement est l'espace commun aux agents. Il est doté en outre d'objets situés passifs sur lesquels les agents peuvent agir : percevoir, détruire, modifier. L'environnement d'un agent est l'environnement du SMA et l'ensemble des agents y évoluant.
- *d'une organisation* : C'est l'agencement des relations, des hiérarchies entre les agents. Dans les systèmes multi-agents, le but peut être de faire apparaître l'organisation.
- *d'un ensemble de points de vue* : Ce sont les descripteurs du système. Ils peuvent être omniscients ou montrer la perception qu'a chaque agent de son environnement.

Des exemples célèbres de système multi-agents sont donnés par les modèles d'insectes sociaux comme les fourmis ou par les modèles de déplacement de troupeau. Pour ce dernier, Reynolds a montré qu'en attribuant à chaque individu d'un troupeau quelques lois de déplacement et d'interaction simples (évitement, convergence, répulsion), on pouvait reproduire les mouvements complexes du groupe entier.

2. PLATE FORMES DE MODELISATION MULTI AGENT

a) La plate forme CORMAS

Il s'agit d'un environnement de développement de systèmes Multi-Agents dédié à la gestion de ressources naturelles développé par le CIRAD [Le Page *et al* 2001]. Il est particulièrement adapté à l'implémentation de modèles d'interactions entre dynamiques naturelles et dynamiques sociales grâce à un ensemble d'objets prédéfinis.

La plate forme CORMAS définit trois catégories d'entités : spatiales, sociales et passives.

- Les entités spatiales constituent l'environnement. Elles possèdent entre autres attributs la collection des cellules voisines, la collection des agents situés sur la cellule...
- Les entités sociales sont les agents proprement dits ; ils peuvent être situés, communicants.
- Les entités passives regroupent les autres objets utilisés dans le modèle : il s'agit des protocoles de communication, des objets situés sur la grille spatiale.

La Figure 6 explicite l'architecture de la plate forme en présentant les relations d'héritage qui relient les différentes classes.

L'implémentation d'un nouveau modèle multi-agent se fait en créant de nouvelles classes qui héritent des classes décrites.

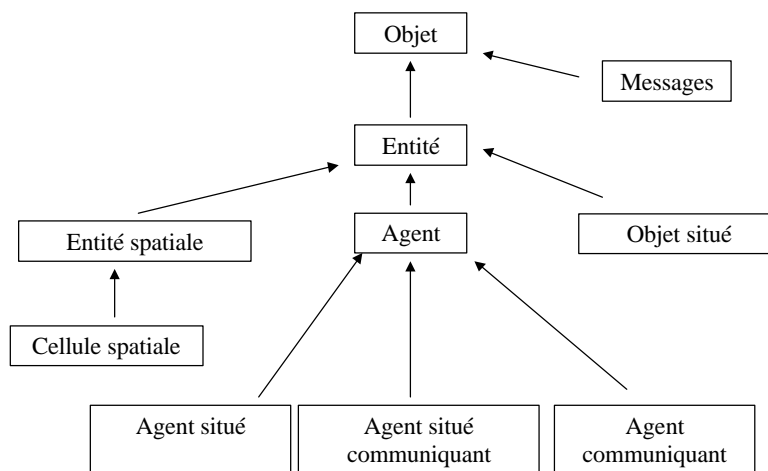


Figure 6 : Architecture de la plate forme CORMAS

b) Exemple d'une application : le modèle Bacsim

Les simulations en écologie visent en général à formaliser la dynamique de populations en interactions entre elles et avec leur milieu. On va dans un modèle multi-agent de dynamique de populations réaliser des simulations "individus centrées", c'est-à-dire que chaque individu est représenté au lieu de travailler sur des groupes et des moyennes. Ce genre de simulations réunit les approches éthologique (science des comportements animaux) et écologique (dynamique des populations) et permet d'inclure la variabilité interindividuelle et les interactions spatiales dans les modèles écologiques. Cette approche aboutit à des modèles biologiquement plus réalistes mais nécessite des temps de calcul très longs.

A l'heure actuelle et à notre connaissance, il n'existe qu'un seul autre modèle multi-agent de biofilm : il a été réalisé par J-U Kreft et G. Booth [Kreft *et al* 2001]. Il s'agit d'un modèle écologique individu centré (IBs) où les agents sont les bactéries elles-mêmes. Il modélise séparément d'une part la croissance et le comportement d'individus bactériens comme agents autonomes, et d'autre part la diffusion-réaction des solutés.

Chaque bactérie est modélisée comme une sphère, la consommation de substrat est décrite par une loi de Monod avec limitations par le substrat et le produit. Les divisions cellulaires suivent le modèle de Donachie [Donachie Robinson 96] qui détermine le volume cellulaire au moment de la division. L'étalement de la biomasse générée suit une règle de distance minimale entre deux cellules voisines, basée sur le calcul du rayon de recouvrement avec chacun des voisins

Les concentrations des solutés sont décrites par une balance massique incluant les phénomènes de transport et de réaction. Les vitesses de réaction sont déterminées par les agents et les concentrations des différents solutés, donc à chaque pas de temps, pour chaque cellule spatiale, on recalcule la vitesse de réaction locale à partir des taux spécifiques de tous les agents de la cellule.

Cette approche a l'intérêt de se baser directement sur la dynamique de chaque bactérie, considérée individuellement. Mais elle nécessite de modéliser un grand nombre d'agents ; les concentrations en bactéries d'un biofilm sont en effet de l'ordre de 10^{11} bactéries par millilitre de biofilm.

B. Description du modèle SMA

1. POSITIONNEMENT ET BUT DU MODELE

Notre but est de mettre au point un modèle qui permette de simuler le développement de biofilms en fonction des conditions appliquées au milieu de développement, et donc par extension en fonction de la loi de commande du réacteur. Ce modèle doit permettre de tester des hypothèses sur la dynamique des biofilms et vérifier des constats expérimentaux.

Dans ce contexte, pourquoi faire appel à la modélisation multi-agents alors que de nombreux modèles ont déjà été réalisés à base d'équations différentielles?

a) Adéquation avec la phase disjointe de développement du biofilm

Le biofilm constitué est une structure où sont associées de manière hétérogène et stratifiée bactéries, matrice extracellulaire et lacunes. Dans ce milieu, la notion de colonie en tant qu'individualités n'est pas très justifiée ; les observations microscopiques ne font pas état de colonies identifiables dans un biofilm formé mûre, du moins dans le cas des biofilms de lits fixes.

Par contre pendant l'initiation d'un biofilm, la notion de colonie est beaucoup plus justifiée ; la colonisation du milieu se fait par taches, des colonies disjointes se fixent et se développent sur la surface. Elles finissent par se rejoindre et former un biofilm mûre. Sur les biofilms de réacteur à lit turbulent, où les contraintes sur le biofilm sont beaucoup plus fortes, on identifie des peuplements par colonie.

Dans un modèle à base d'équations, la phase disjointe du biofilm est dure à décrire alors qu'elle est naturelle pour un SMA.

b) Adéquation avec le point de vue biologiste

La réalisation d'un système multi-agent pour décrire un biofilm revient à décrire un système macroscopique par ses constituants microscopiques, d'origine biologique. Le fonctionnement d'un modèle basé sur les connaissances sur les micro-organismes, et sa fidélité à la réalité, a l'avantage de tester la validité de ses connaissances et met en évidence les phénomènes pour lesquels nos connaissances sont insuffisantes.

Les études sur les biofilms ont pu montrer que son historique déterminait pour une part non négligeable les réactions d'un biofilm aux modifications des conditions du milieu [Michaud 2001]. Cet enregistrement du passé se fait sous la forme de production de matrice extracellulaire, d'orientation métabolique, ... Le fait de définir des agents, c'est à dire des individus, dans le biofilm permet de fait de les doter de « mémoire ».

Un autre intérêt des agents est qu'ils sont potentiellement différents et peuvent donc exprimer la variabilité génétique des individus dans une population. Enfin, ils se prêtent tout à fait à l'utilisation de modèles génétiques, où chaque agent est doté de chromosomes qui décriraient ses voies métaboliques et les constantes métaboliques associées.

c) Description du modèle proposé

Conformément à l'archétype d'un SMA, nous avons trois entités à considérer : l'espace, les agents et l'observateur.

Le modèle proposé définit les agents comme des groupes de bactéries, assimilés à des colonies. Ces agents sont localisés et forment le biofilm. Il est à noter que la définition des agents utilisée est finalement assez abstraite et ne repose pas sur des individualités très marquées. La définition d'un agent « bactérie » impose des capacités de calcul importantes du fait du nombre d'agents qu'il faudrait simuler. Ce problème est permanent en modélisation écologique individu-centrée, où le nombre d'individus à simuler peut vite devenir très grand. Cependant pour des bactéries, où de nombreux individus sont identiques, il est peut-être possible de définir un agent abstrait, qui serait un groupe de bactéries, de masse variable et d'effectif inconnu, et qui identifierait les caractéristiques et les réactions d'agents biologiques. Par rapport à un agent cellulaire, un certain nombre de comportements ne sont plus appropriés : division cellulaire, pression mécanique entre les cellules. Il faut en fait considérer cet agent comme une masse réactante, sur laquelle nous pouvons appliquer les techniques classiques de balance massique.

Ce choix d'un agent virtuel est la principale différence avec le modèle de Kreft. Il doit permettre d'effectuer des simulations à plus grande échelle, puisque nous avons moins d'agent à modéliser ; le modèle réalisé devrait ainsi être plus facile à relier au fonctionnement général du réacteur.

De fait l'espace est divisé en deux parties : la phase libre, qui ne contient aucun agent mais n'est pas réactive et la phase biofilm.

- La phase libre est supposée parfaitement remuée et soumise à l'alimentation du réacteur. Les concentrations sont donc constantes en tout point de la phase libre.
- La phase biofilm n'est pas soumise aux mouvements de convection, les concentrations sont fixées par la diffusion des molécules et les réactions biochimiques.

- Une troisième phase a été ajoutée, c'est la couche limite entre le biofilm et la phase libre ; cette couche d'épaisseur variable selon les conditions d'alimentation et le type de réacteur, n'est pas réactive mais n'est pas soumise au mélange : elle suit donc un régime diffusif.

Le système modélisé n'est pas seulement le biofilm, mais le biofilm et la solution du réacteur. Les modélisations de biofilm considèrent souvent les propriétés de la phase liquide à proximité du biofilm comme constantes ; c'est l'hypothèse d'un réacteur parfaitement mélangé et contrôlé. Nous avons fait le choix de tenir compte des interactions biofilm – liquide. Il reste à déterminer si l'échelle de modélisation permet une généralisation suffisante pour prévoir l'impact des stratégies d'alimentation sur le biofilm.

2. LES DIFFERENTS PHENOMENES A MODELISER

Nous avons adopté l'hypothèse classique en modélisation de biofilm, selon laquelle deux dynamiques d'échelle de temps différentes décrivent le biofilm. Ainsi la modélisation de l'environnement tel que nous l'avons définie correspond à la dynamique rapide du biofilm (équilibre atteint en quelques secondes), et la modélisation des agents correspond à la dynamique lente (équilibre en quelques heures voire quelques jours).

La modélisation métabolique choisie est une version inspirée de celle qui a été fixée pour le modèle AMOCO : seules deux réactions successives sont considérées, la réaction acétogène et la réaction méthanogène.



Un pas de modélisation de l'environnement doit traiter les phénomènes de

- Diffusion des différents solutés
- Convection, contrainte hydrodynamique
- Réactions chimiques, équilibres acido-basiques,
- Réactions biochimiques (en interaction avec les agents)

Seules les réactions biochimiques ont été considérées, les équilibres acido-basiques et gazeux n'étant pas indispensables au fonctionnement du modèle. Cependant, l'architecture actuelle permet d'ajouter les calculs de ces réactions sans difficultés. De même les phénomènes de convection n'ont pas été retenus. Ceci est dû à la définition des agents et du biofilm : le biofilm est défini par la présence d'agents. Les lacunes et canaux du biofilm n'étant pas représentés, il est impossible de définir de convection. Nous avons fait l'hypothèse que l'utilisation de coefficients de diffusivité assez élevés permet de rendre compte macroscopiquement des phénomènes de diffusion-convection observés microscopiquement.

Un pas de modélisation d'un agent doit réaliser les opérations suivantes :

- Loi de croissance - mort
- Loi d'interaction physique, colonisation

a) Croissance cellulaire

La vitesse de croissance des micro-organismes est classiquement proportionnelle à leur concentration. Le coefficient de proportionnalité, noté μ et appelé taux spécifique de croissance, est une fonction des conditions du milieu. Il existe de nombreux modèles pour calculer ce taux de croissance dont la plupart est basée sur le modèle de Monod (1942) qui relie μ à la concentration des molécules limitantes (substrats, inhibiteurs).

$$\frac{dX}{dt} = \mu X = \mu_{\max} \frac{S}{S + K_s} X \quad (1)$$

μ_{\max} (j^{-1}) est le taux de croissance maximal observé à saturation en substrat, K_s (g/L) est la constante de demi-saturation, qui correspond à la concentration pour laquelle $\mu = \mu_{\max}/2$.

Il existe de nombreuses variantes du modèle de Monod, permettant de rendre compte d'inhibitions par le substrat ou par un produit de la réaction.

La relation (1) décrit la croissance en terme d'augmentation de concentration en micro-organisme. Il n'y a pas de modèle général qui relie la croissance du biofilm à la croissance en concentration. Il faudra donc établir un modèle de croissance en épaisseur du biofilm, compatible avec les données expérimentales.

Dans le cas particulier de notre exemple d'étude, les propriétés cinétiques suivantes ont été déterminées à l'échelle du réacteur à partir des mesures effectuées à l'équilibre pour une série de conditions d'alimentation [**Bernard et al 2001**] :

$$\begin{cases} \mu_1 = \mu_1^{\max} \frac{S_1}{S_1 + K_1^s} \\ \mu_2 = \mu_2^{\max} \frac{S_2}{S_2 + K_2^s + \left(\frac{S_2}{K_2^I} \right)^2} \end{cases}$$

acétogénèse	μ_1^{\max}	1,2	jour ⁻¹
	K_1^s	7,1	g.L ⁻¹
méthanogénèse	μ_2^{\max}	0,74	jour ⁻¹
	K_2^s	0,55	g.L ⁻¹
	K_2^I	0,96	g.L ⁻¹

Généralement, l'inhibition des acétogènes n'est pas apparente car celle des méthanogènes apparaît avant. Ces paramètres ont été déterminés macroscopiquement et ne sont peut-être pas vérifiés microscopiquement. Néanmoins, ce sont ceux que nous utiliserons dans un premier temps afin de conduire les tests préliminaires du modèle avant de les identifier plus précisément.

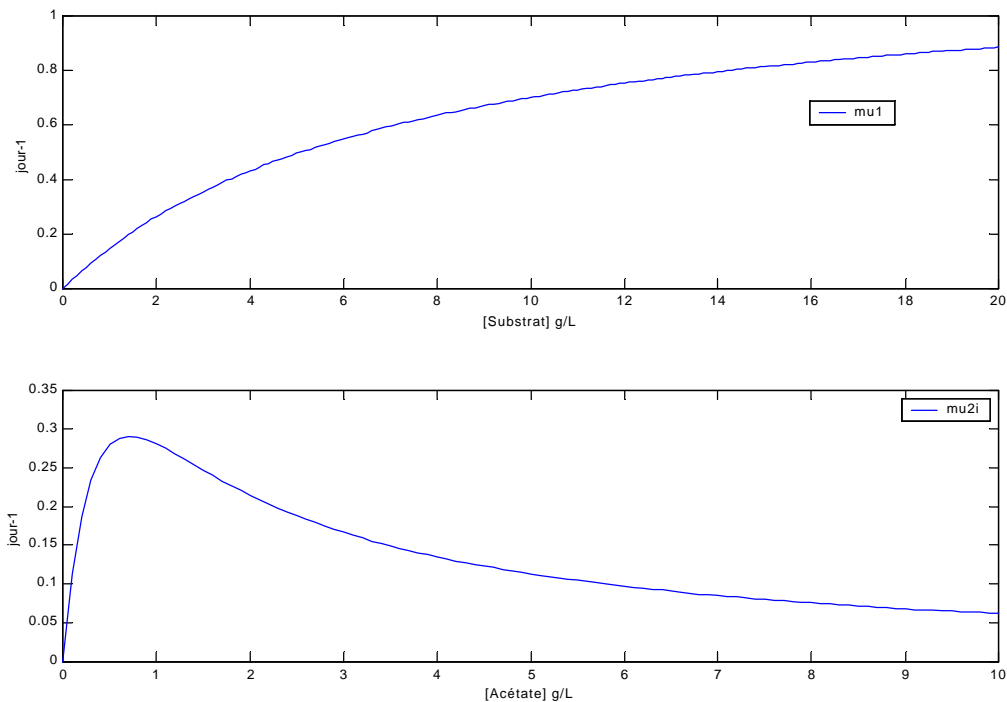


Figure 7 : Evolution des taux spécifiques acétogènes (μ_1) et méthanogènes (μ_2)

b) Disparition de biomasse

La disparition nette de biomasse active regroupe les phénomènes d'arrachement de biofilm, de mort bactérienne, et d'inactivation. Les observations expérimentales montrent que l'arrachement est fonction de l'activité des bactéries et du débit imposé.

- Lorsque le débit augmente, les forces d'abrasion aussi. Selon le volume du réacteur, le débit d'alimentation ne représente pas la même force d'abrasion ; on préférera donc relier les forces d'abrasion au temps de séjour hydraulique (TSH), défini par le rapport :

$$TSH = \frac{V_{utile}}{Débit}$$

- Une augmentation de la concentration en substrat crée une augmentation de l'activité des bactéries, la vitesse de division s'accélère mais la matrice extracellulaire n'est pas produite en conséquence, ce qui facilite donc l'arrachement des bactéries. Dans ce cas, le terme de croissance nette reste positif.

La valeur moyenne du terme d'arrachement γ est identifiée à partir des conditions d'équilibre issues du modèle global. A l'équilibre, les biomasses sont constantes, ce qui se traduit macroscopiquement par $\gamma^{obs} = \mu^{obs}$. Cependant l'arrachement n'est pas identique au sein du biofilm, des bactéries proches de la périphérie sont plus facilement arrachables.

La relation $\gamma=\gamma(x)$ doit être telle que la valeur moyenne soit égale à la valeur observée, ce qui

se traduit par :
$$\int_0^{x_{\max}} g dx = g_{obs} * (x_{\max} - 0) .$$

La formule décrivant le terme d'arrachement doit être croissante en fonction de l'épaisseur et décroissante en fonction du temps de séjour. On suppose dans un premier temps une relation linéaire

$$g(x) = \frac{a}{TSH} x ,$$
 et on identifie le gain grâce à l'intégrale. On en déduit donc

$$g(x) = \frac{2g_{obs}}{TSH} \frac{x}{(e_{obs})} \tag{2}$$

où e_{obs} est l'épaisseur du biofilm correspondant au γ_{obs} .

c) Distribution de la biomasse

La production nette de biomasse d'une colonie

est $\frac{dX}{dt} = (m-g)X$. Lorsque ce terme est positif, il sert soit à la croissance de la colonie, soit à la création et au développement de nouvelles colonies dans le voisinage.

Les règles de distribution de biomasse sont telles que :

- la masse m d'une colonie ne peut dépasser une valeur seuil,
- la densité de biomasse d'une cellule spatiale ne peut dépasser une valeur seuil
- le nombre de colonies présentes sur une même cellule est limité.

Une série de tests sur la variation de biomasse (dm), la masse de l'agent, la biomasse totale présente sur la cellule spatiale, et l'occupation du voisinage permet de déterminer la distribution de la biomasse à chaque pas de temps en respectant les conditions décrites ci-dessus.

La Figure 8 décrit l'enchaînement des tests pour la prise de décision pour la distribution de biomasse.

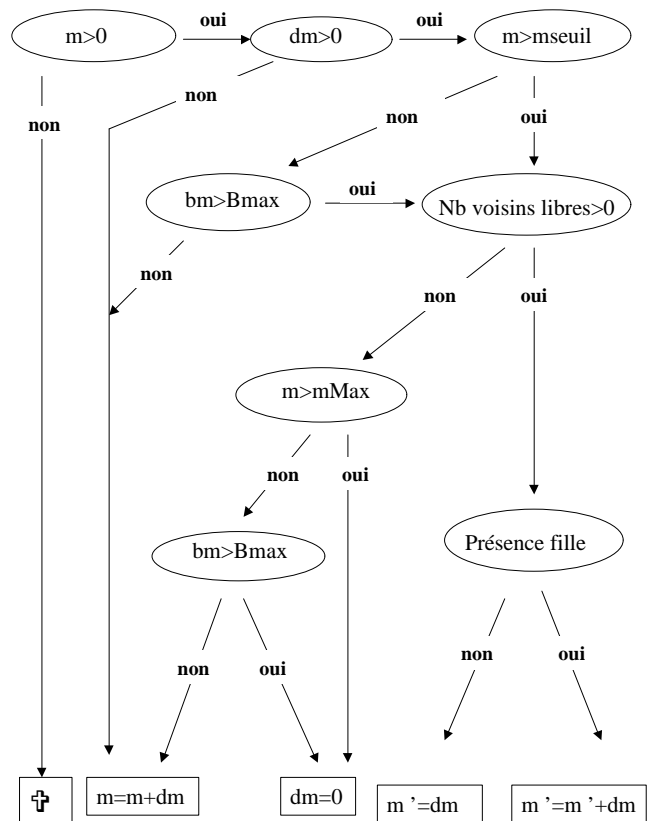


Figure 8 : Diagramme de décision pour la distribution de biomasse

d) Réaction

La cinétique des réactions biochimiques suit, elle aussi, la loi de Monod, à un terme de rendement près.

$$\left(\frac{dS}{dt}\right)_{Reac} = k_s mX \quad (3)$$

Espèce chimique	Bactérie effectrice	Constante	Valeur (g/gXi)
Substrat (S1)	Acétogène (X1)	k11	-42
Acétate (S2)	Acétogène (X1)	k12	6,99
Acétate (S2)	Méthanogène (X2)	k21	-16,09
Méthane (S3)	Méthanogène (X2)	k22	5,04

e) Diffusion

Le biofilm et la couche limite sont des milieux où les solutés sont soumis entre autres phénomènes à la diffusion.

La diffusion suit la loi de Fick : $J = -D_s \frac{\partial S}{\partial x}$

où J est le flux massique par unité de surface, D la diffusivité en m²/j, S la concentration en g/L. Un bilan massique sur un volume élémentaire

donne : $dm = A(J(x) - J(x + dx))dt$

d'où $dS = \frac{dm}{A dx} = \frac{J(x) - J(x + dx)}{dx} dt$

On en déduit que la variation de concentration due à la diffusion d'un soluté est

$$\left(\frac{dS}{dt}\right)_{diff} = D_s \frac{d^2 S}{dx^2} \quad (4)$$

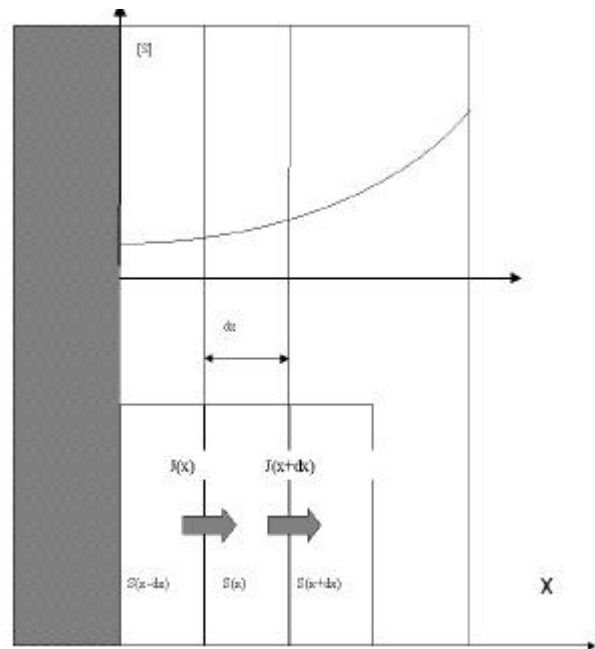


Figure 9 : Illustration de la loi de Fick

Les valeurs de diffusivité sont données pour des diffusions dans l'eau. Par rapport à de l'eau pure, un biofilm va présenter une résistance à la diffusion : les valeurs de diffusivité classiquement rencontrées dans les biofilms varient d'un facteur 0,3 à 0,8 par rapport à la diffusivité de l'eau. Nous avons appliqué une relation affine reliant la diffusivité à la densité de biomasse.

3. INTERACTION GLOBAL / LOCAL

a) De l'alimentation à la phase liquide

En permanence un débit d'entrée est appliqué au réacteur ; le volume du réacteur étant constant, le même débit sort du réacteur. Le débit d'entrée est caractérisé par ses concentrations d'entrée en substrat et en acétate, et par sa valeur en L/h.

$$V_u \frac{dS^\circ}{dt} = Q_{in} S_{in} - Q_{out} S^\circ + Pr od - Conso$$

En appliquant l'hypothèse que les débits d'entrée et de sortie sont égaux (régime permanent) et en exprimant le débit en fonction du Temps de séjour, on obtient la relation

$$\frac{dS^\circ}{dt} = \frac{S_{in} - S^\circ}{TSH} + \frac{Pr od - Conso}{V_u}$$

b) Du biofilm à la phase liquide

Nous faisons maintenant l'hypothèse que les réactions chimiques n'ont lieu que dans le biofilm. Le terme Production - Consommation est donc égal au flux de matière observé à la couche limite du biofilm.

Le flux est décrit par la loi de Fick (cf. § précédent), En s'appuyant sur l'hypothèse simplificatrice que le flux observé au niveau du biofilm est normal à sa surface, on ne dérive donc que selon l'axe x. Le flux de matière (g/m²) est rapporté au réacteur par sa surface spécifique σ exprimée en m²/m³.

$$S = \frac{Surface_{biofilm}}{Volume_u}$$

$$\frac{dS^\circ}{dt} = \frac{S_{in} - S^\circ}{TSH} + JS$$

On obtient finalement la relation :

$$\begin{aligned} \frac{dS^\circ}{dt} &= \frac{S_{in} - S^\circ}{TSH} - S D_s \left(\frac{dS}{dx} \right)_{BL} \\ \frac{dAc^\circ}{dt} &= \frac{Ac_{in} - Ac^\circ}{TSH} - S D_{Ac} \left(\frac{dAc}{dx} \right)_{BL} \end{aligned} \quad (5)$$

c) Expression des paramètres d'état

Le fonctionnement des réacteurs peut être suivi à partir de plusieurs capteurs en ligne. Les capteurs sont souvent utilisés pour quantifier l'entrée et la sortie du réacteur mais rarement sur l'intérieur même du réacteur.

Afin de pouvoir comparer au mieux les résultats du modèle et les données expérimentales, nous avons calculé les variables du modèle « vues de l'extérieur ».

Les concentrations de sortie en substrat et en acétate sont celles de la phase libre.

Débit de méthane

Le biofilm produit à chaque pas de temps $m_{CH_4} = dl^3 \sum S_3$. En ramenant cette production au volume modélisé et au temps, on obtient l'expression du débit massique en $g.L^{-1}.j^{-1}$

$$q_{CH_4} = S \frac{dl^2}{h.Dt} \sum S_3 \quad (6)$$

Concentration en biomasse

On ramène la masse de biomasse présente dans la portion de biofilm modélisé au volume modélisé :

$$X_g = S \frac{dl^2}{h} \sum X \quad (7)$$

4. ARCHITECTURE DU MODELE

a) Diagrammes du modèle multi-agents

Le diagramme suivant montre l'architecture du modèle en terme et son intégration dans la plate forme CORMAS. Il indique pour chaque classe du modèle, la classe dont elle hérite, ses variables et ses méthodes.

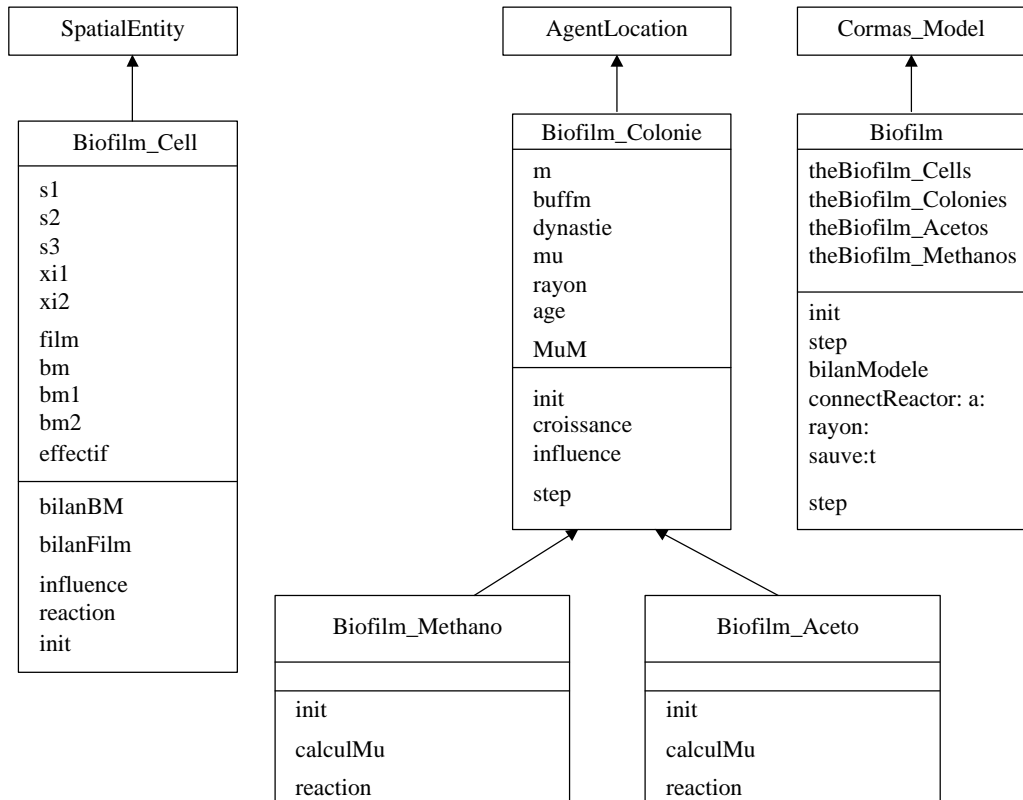


Figure 10 : Diagramme de classes du modèle

Le modèle comprend cinq classes :

- **Biofilm_Cell** qui est un élément de l'environnement discret ; elle hérite de la classe *SpatialEntityElement* de Cormas.
- **Biofilm_Colonie** qui représente une colonie quelconque.
- **Biofilm_Aceto** qui représente une colonie acétogène
- **Biofilm_Methano** qui représente une bactérie méthanogène.
- **Biofilm** qui initie et conduit la simulation

Biofilm_Methano et **Biofilm_Aceto** héritent d'une même classe **Biofilm_Colonie** qui possède les attributs et les méthodes communes aux colonies. **Biofilm_Colonie** est un agent localisé, il hérite de la classe **AgentLocation** définie dans Cormas.

L'objet **Biofilm_Cell** décrit la dynamique de l'environnement : évolution des concentrations, appartenance au biofilm.

Les objets **Biofilm_Colonie** sont des agents qui évoluent dans l'environnement : ils sont fixes mais peuvent croître, se réduire, se multiplier ou disparaître. Ils sont *a priori* indépendants, mais peuvent posséder une descendance. Ils diffèrent entre eux par leur appartenance à une population (acétogène ou méthanogène), leur masse, leurs constantes métaboliques.

L'environnement et les agents interagissent fortement : les solutés présents dans l'environnement déterminent la dynamique des agents ; l'objet **Biofilm_Cell** réalise la réaction, mais en se servant des paramètres métaboliques des agents locaux.

La procédure d'initialisation est effectuée une fois, la méthode **Biofilm step** est exécutée à chaque pas de temps.

La procédure d'initialisation affecte les valeurs initiales de concentration des solutés de l'espace (**Biofilm initCells**) puis crée et place les agents sur la grille (**Biofilm initAgents**), en leur attribuant une masse et leurs propriétés métaboliques.

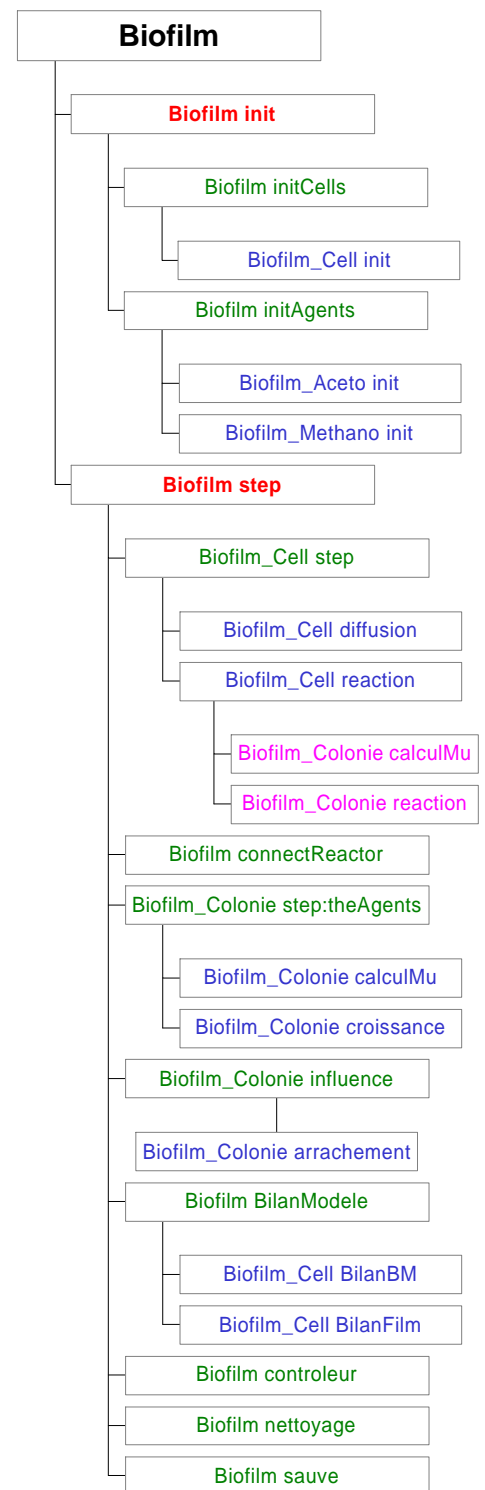


Figure 11 : Diagramme d'exécution du modèle

A chaque pas de temps on effectue plusieurs fois le pas de simulation de l'environnement (**Biofilm_Cell step**) et une fois le pas de simulation des agents (**Biofilm_Colonie step**).

La méthode **Biofilm_Cell** step appelle la méthode **Biofilm_Cell Diffusion** qui effectue la diffusion des solutés à partir d'une loi de Fick discrétisée, puis la méthode **Biofilm_Cell Reaction**.

Cette dernière appelle la méthode **calcul_mu** des agents de type **Biofilm_Aceto** et **Biofilm_Methano** afin de déterminer le taux spécifique observé pour la cellule, et effectue le calcul des réactions selon la loi de Monod.

Les changements de concentration dus à la diffusion et à la réaction sont sommés dans une variable de la cellule. A la fin du pas de temps, cette variable représente la variation de concentration de la cellule.

La méthode **Biofilm_Cell** influence met à jour les concentrations en fonction de ces variables de variation. En procédant ainsi, tous les calculs effectués successivement pour un même pas de temps utilisent les mêmes valeurs numériques, on respecte la simultanéité des phénomènes malgré un calcul séquentiel.

La méthode **Biofilm connectReactor** met à jour les concentrations des solutés dans la phase libre en fonction de l'alimentation et des réactions

Il est important de souligner que bien que les calculs soient séquentiels, les phénomènes restent synchronisés au sein de chaque dynamique. En effet, les méthodes **diffusion** et **réaction** pour les cellules, **croissance** et arrachement pour les agents ne modifient pas directement la variable qui les concerne mais ajoutent leur "influence" sur la variable dans une variable tampon. A la fin de chaque cycle, les méthodes **Biofilm_Cell influence** et **Biofilm_colonie influence** modifient la variable en fonction du buffer.

b) Constantes et paramètres

Le tableau suivant regroupe toutes les constantes et les paramètres du modèle.

	Paramètre	Unité	Valeur usuelle
Pas de temps	Dt	Jour	0,001
Pas spatial	Dl	m	50.10^{-6}
Concentration initiale en substrat	S0	g/L	2
Concentration initiale en acétate	Ac0	g/L	0,7
Alimentation en substrat	Sin	g/L	10
Alimentation en équivalent acétate	Acin	g/L	2
Temps de séjour hydraulique	TSH	Jour	1
Surface spécifique	Sigma	m^2/m^3	180
Concentration initiale en bactérie acétogène	X1°	g/L	5
Concentration initiale en bactérie méthanogène	X2°	g/L	5
	Epaisseur	m	400.10^{-6}
Détachement de biomasse observé	Gamma obs	jour ⁻¹	0,4
Epaisseur du biofilm correspondant	Eobs	m	100.10^{-6}
Diffusivité du substrat	D1	$m^2/jour$	$0,6.10^{-4}$
Diffusivité de l'acétate	D2	$m^2/jour$	$1,03.10^{-4}$
Concentration maximale dans le biofilm	BMax	g/L	30
Seuil de colonisation	Bseuil	g/L	1

5. INDICATEURS DEVELOPPES

Une série d'indicateurs ont été développés afin de pouvoir analyser les résultats de simulation. Des fichiers de sauvegarde sont créés tous les 50 pas de temps (ce qui correspond à 1h12 pour un pas de temps de 0.001 jours). Ils décrivent l'état de l'environnement (concentrations des solutés, densité des biomasses) en tout point de l'espace modélisé, la répartition en masse et en âge des agents, ainsi que les paramètres d'état du réacteur (alimentation, concentration des solutés dans la phase libre, débit de méthane, concentration globale de biomasse).

Les fichiers sont traités par Matlab pour fournir l'évolution des profils de concentration, de densité et d'activité au sein du biofilm, l'évolution du biofilm (surface, périmètre), l'évolution du réacteur.

C. Adaptations préliminaires

1. QUEL ALGORITHME DE DIFFUSION - REACTION

La modélisation d'un biofilm comprend très souvent la modélisation du phénomène physico-chimique de diffusion des molécules et de leur réaction chimique. Certains modèles supposent qu'il n'y a pas de gradient de solutés au sein du biofilm et se contentent de modéliser les réactions chimiques et biochimiques. Cette hypothèse est vérifiée dans le cas des biofilms fins (moins de 100 μm). Dans notre cas, nous voulions d'une part établir un modèle le plus générique possible et d'autre part le valider sur l'exemple des lits fixes où l'épaisseur du biofilm est suffisante pour que l'effet de la diffusion ne soit pas négligé.

a) Les différents algorithmes

La diffusion suit une loi de Fick, la réaction biochimique suit une loi de Monod. En combinant ces deux relations, et en faisant l'hypothèse qu'il n'y a pas de variation de la concentration S selon l'axe y, on obtient pour un biofilm à l'équilibre l'équation différentielle suivante,

$$-\frac{dJ}{dx} + kX = 0$$

ce qui se traduit pour une équation de Monod du premier degré par l'équation différentielle :

$$D \frac{d^2 S}{dx^2} + \frac{kXS}{K} = 0 \quad (8)$$

Cette relation est complétée par les conditions aux limites : la première traduit la continuité avec la phase libre, la deuxième traduit le fait qu'aucun flux de matière ne puisse traverser le support.

$$S|_{x=L} = S^{\circ} \text{ et } \left. \frac{dS}{dX} \right|_{x=0} = 0$$

Cette équation a une solution analytique [Droste 86] :

$$S = S^{\circ} \frac{\cosh\left(\frac{xf}{L}\right)}{\cosh(f)} \text{ où } f = L \sqrt{\frac{kX}{DK}}$$

Ceci est la solution pour une réaction qui suit une cinétique du premier degré. Dans le cas de réactions en chaînes et aux cinétiques plus complexes, il n'est pas possible de fournir de solutions analytiques. On fait alors appel à des techniques numériques qui par calculs itératifs permettent de fournir une solution au problème.

En appliquant une discrétisation spatiale et temporelle à la relation (8), on obtient en monodimensionnel pour chaque maille 'j' de l'environnement

$$\left. \frac{dS}{dt} \right|_j = \left(\frac{D}{dl^2} \frac{d^2 S}{(dX)^2} + kmX \right) dt$$

soit :

$$S_x^{t+dt} = S_x^t + \frac{D}{dl^2} (S_{x+1}^t + S_{x-1}^t - S_x^t) dt = S_x^t (1 - 2D \frac{dt}{dl^2}) + S_{x-1}^t + S_{x+1}^t$$

La dernière formulation montre que le pas de temps doit être choisi en fonction de la largeur de maille choisie. En effet si le rapport dt/dl^2 est trop élevé, on risque de « faire diffuser » plus de matière qu'il n'y en a dans la maille et on aboutit alors à des concentrations négatives.

En appliquant cette relation en deux dimensions, on peut alors calculer pour chaque pas de temps la concentration en chaque maille du plan.

$$S_{x,y}^{t+dt} = S_{x,y}^t + D \frac{dt}{dl^2} \left[(S_{x-1,y}^t + S_{x+1,y}^t - 2S_{x,y}^t) + (S_{x,y-1}^t + S_{x,y+1}^t - 2S_{x,y}^t) \right]$$

La méthode Biofilm diffusion utilise cette relation pour calculer à chaque pas de temps la variation de concentration de chaque soluté pour chaque cellule spatiale. Cependant on est obligé d'utiliser un pas de temps très petit au vu des largeurs de mailles et des coefficients de diffusivité utilisés. A titre d'exemple, il faut un pas de temps de 340 millisecondes pour une largeur de maille de 50 μm mais de 34 millisecondes pour une largeur de maille de 10 μm .

Or les résolutions habituellement utilisées sont de l'ordre du μm . Cette méthode de résolution n'est donc pas satisfaisante. La méthode des **D**irections **A**lternatives **I**mplicites de Peaceman et Rachford (**ADI**) permet d'effectuer le calcul des problèmes multi-dimensionnel de diffusion-réaction (cf. en annexe). Elle a l'avantage de ne pas nécessiter d'hypothèses d'état stationnaire et d'offrir une relative simplicité calculatoire (des systèmes d'équations en série). Elle repose sur deux équations de différences finies utilisées alternativement par demi pas de temps qu'on balaye sur le plan selon les deux axes alternativement. Elle génère alors des systèmes d'équations triangulaires.

Afin de résoudre les systèmes d'équation, il convient d'ajouter des conditions aux bordures. En plus des conditions aux bordures classiques présentées dans la résolution adimensionnelle (continuité avec la phase libre et dérivée nulle sur le support), on ajoute un bouclage de l'espace selon l'axe y : ceci revient à supposer qu'il n'y ait pas de gradient selon y. Cette hypothèse est valable pour des petits volumes de simulation.

La résolution de la réaction diffusion par la méthode ADI demande la résolution, pour un plan de N colonnes et L lignes, de N systèmes d'équations à L lignes et L systèmes à N lignes, et ce pour chaque soluté, à chaque pas de temps.

b) Optimisation du pas de temps

Lorsqu'on effectue un calcul de réaction-diffusion, le système arrive souvent à une situation d'équilibre, les transferts locaux de masse deviennent alors très faibles. Il est alors inutile d'utiliser des pas de temps aussi fins que lorsqu'on était loin d'une situation d'équilibre.

C'est pourquoi nous avons ajouté à la boucle de calcul, une mise à jour du pas de temps en fonction de l'écart à la position d'équilibre. Plusieurs indicateurs ont été testés, basés soit sur l'écart-type, soit sur le transfert de matière.

Un contrôle visant à maintenir un transfert de biomasse par pas de temps supérieur à un seuil s'est montré efficace sans perturbation significative du résultat final.

c) La passerelle Matlab

L'algorithme de résolution a d'abord été développé sous Matlab ; cet algorithme utilisant essentiellement des calculs matriciels pour lesquels Matlab est efficace, nous avons décidé d'utiliser une passerelle entre le modèle multi-agent proprement dit (qui est programmé en Small Talk) et la résolution du champ de concentration.

La passerelle consiste en l'établissement d'un lien de type **Dynamic Data Exchange** entre VisualWorks et Matlab. Les liaisons DDE unissent une application *cliente* qui initie la conversation, et une application *serveur* qui effectue des tâches pour l'application cliente. Ce type de lien nécessite que les deux applications soient ouvertes.

La classe de communication a été développée à partir de celle réalisée pour le couplage Visual Works - Vensim par Eric Piquet et Christophe Le Page.

Le type de donnée échangé est au format texte ; dans ce format, il est apparu que Visual Works n'était capable d'exporter que 800 valeurs numériques. L'algorithme de réaction diffusion a besoin de 4 valeurs numériques par cellule (substrat, acétate, biomasse acétogène, biomasse méthanogène). On est donc ramené à des grilles d'au maximum 200 cases.

L'intérêt de la passerelle est alors assez diminué, mais on a maintenant la possibilité de modéliser sur une gamme d'échelle plus grande sans toutefois augmenter significativement la résolution (nombre de cellules/m) des simulations.

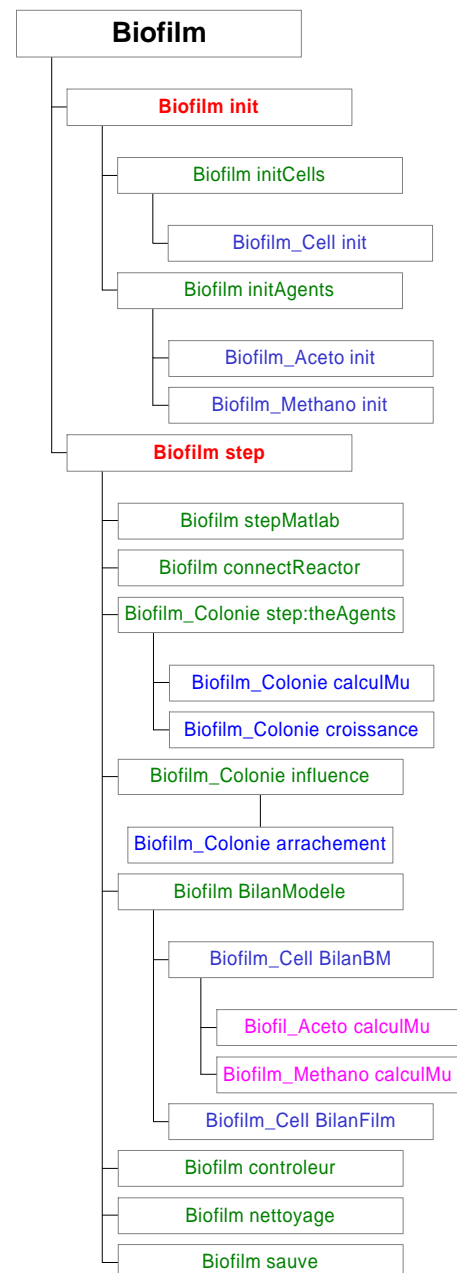


Figure 12 : Diagramme d'exécution du modèle avec passerelle

Il faut noter que le couplage par DDE est instable et limité ; il nécessite de plus que les deux applications soient installées dans leur intégralité et ouvertes, avec tout ce que ça implique en terme d'occupation de la RAM. L'utilisation de DLL (**D**ynamic **L**ink **L**ibraries) semble plus indiquée, il faudrait pour cela compiler l'algorithme Matlab en C++ sous forme de dll.

La passerelle se substitue aux deux méthodes **Biofilm_Cell step** et **Biofilm_Cell influence**. Elle nécessite par contre le calcul des taux spécifiques, inséré dans la méthode **Biofilm_Cell bilanBM**.

2. DESYNCHRONISATION DES AGENTS

A l'initialisation de la simulation, les agents sont créés et placés sur la grille. Si les agents sont tous identiques, on peut faire apparaître des artefacts de simulation comme la division synchronisée de toutes les colonies. Ces synchronisations ne sont pas dues à la simulation d'un phénomène mais au mode de modélisation. On peut éviter ce problème en appliquant un tirage aléatoire pour les propriétés de chaque agent (biomasse initiale, constante de réaction).

3. AJOUT D'UN CONTROLEUR

Notre but est de voir l'effet d'un mode de conduite de réacteur sur le biofilm ; nous avons donc ajouté une méthode *Biofilm contrôleur* qui permet à chaque pas de temps de modifier les paramètres d'alimentation. Dans un premier temps, l'alimentation est choisie constante, le temps de valider le modèle ; le contrôleur se contente de jouer sur le débit en cas de lessivage du réacteur (si la concentration du milieu est égale à la concentration d'alimentation, le contrôleur réduit le débit) ou en cas d'alimentation insuffisante (si la concentration en substrat du milieu est inférieure à 1 g/L on augmente le débit)

Conclusion

L'élaboration du modèle a essayé de répondre aux deux impératifs d'évolutivité et de mise en œuvre calculatoire. Tel qu'il est actuellement, il est possible de rajouter au modèle de nouveaux éléments, de nouvelles interactions, de changer les lois de fonctionnement.

Le modèle s'appuie sur une série d'hypothèses :

- Possibilité de découpler la simulation des dynamiques rapides et lentes du biofilm
- Possibilité de définir des agents non observés mais identifiant les propriétés d'agents plus petits et plus nombreux
- Le flux de molécules est décrit par la loi de Fick sur la diffusion, et le flux sortant est normal à la surface du biofilm
- Les réactions biochimiques sont localisées dans le biofilm
- La biomasse est fixée et ne se déplace pas au sein du biofilm

V. Validation – Evolution du modèle

Un modèle est une simplification de la réalité, il vise à reproduire le comportement d'un système. Pour s'assurer de la validité du modèle, nous pouvons comparer des données réelles aux données prédites par le modèle, mais c'est difficile à réaliser pour les modèles de vie artificielle.

Une première validation consiste à vérifier que le modèle n'aboutit pas à des résultats aberrants mais au contraire conformes à l'expérience qu'on a du système réel. L'étude de sensibilité permet de détecter des biais du modèle et de détecter les paramètres fondamentaux du modèle.

A.	Analyse des résultats _____	35
B.	Etude de sensibilité _____	39
1.	Conduite _____	39
2.	Sensibilité des paramètres de modélisation _____	39
3.	Sensibilité à la séquence des calculs _____	41
C.	Développements ultérieurs du modèle _____	42
1.	Développements du modèle _____	42
2.	Limites de la plate-forme utilisée _____	44

A. Analyse des résultats

Les résultats présentés sont les résultats de deux simulations. La première est une simulation de 15 jours sur 15000 pas, sur un espace de 5×15 cellules avec une maille de $50 \mu\text{m}$. Le système est initialisé avec un biofilm homogène de $100 \mu\text{m}$. Les concentrations d'alimentation sont celles communément rencontrées dans l'effluent du réacteur pilote. Les concentrations initiales des solutés sont celles d'un système à l'équilibre; elles ont été établies à partir du modèle AMOCO qui fournit également les constantes métaboliques du modèle BIOFILM. La deuxième simulation est celle d'un biofilm installé, relativement épais ($500 \mu\text{m}$), alimenté uniquement en substrat.

a) Profils de concentration des solutés

Comme on peut le voir sur la Figure 13, les profils des solutés sont pratiquement plats. Or les constatations expérimentales montrent que les gradients de solutés existent, et peuvent même être parfois très forts. Cet écart peut être interprété soit par l'utilisation de valeurs de diffusivité trop fortes, soit par la surestimation du mélange de la phase libre située en face du biofilm. L'apport d'acides gras volatils dans le système par l'alimentation du réacteur participe aussi à ce nivellement des profils.

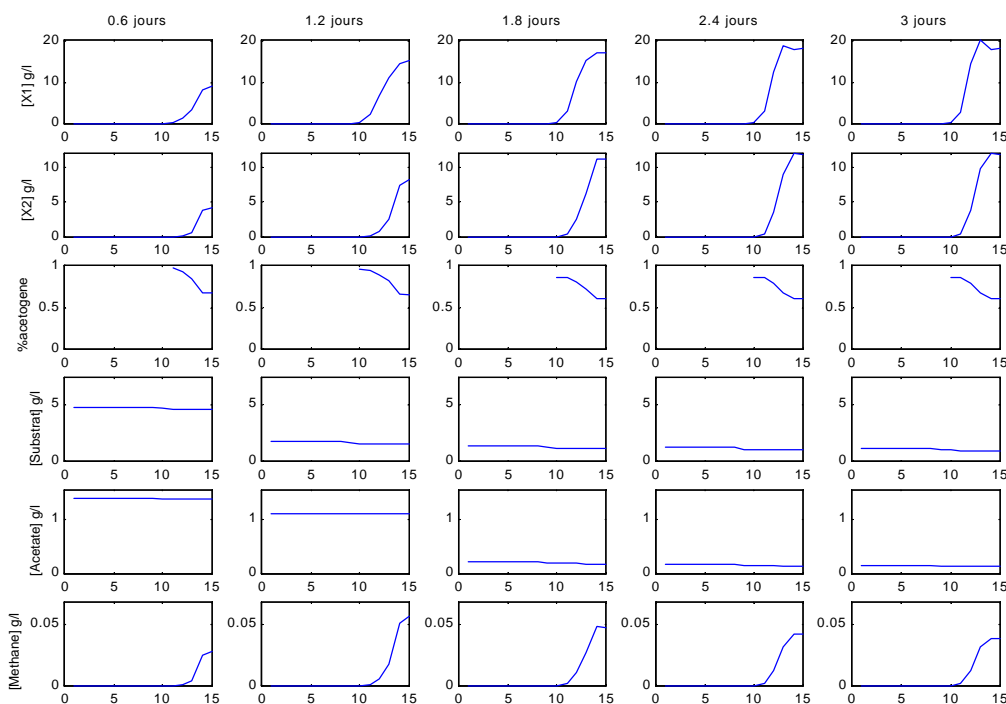


Figure 13 : Profil de concentrations Simulation 1

Dans le cas de la simulation 2 (Figure 14), la suppression d'alimentation en acides gras a pour conséquence la chute de la concentration moyenne du réacteur en acides gras; cette espèce n'est pratiquement plus rencontrée dans la phase libre. Parallèlement un gradient de substrat et d'acétate s'établit, le biofilm entre en contrôle diffusif partiel.

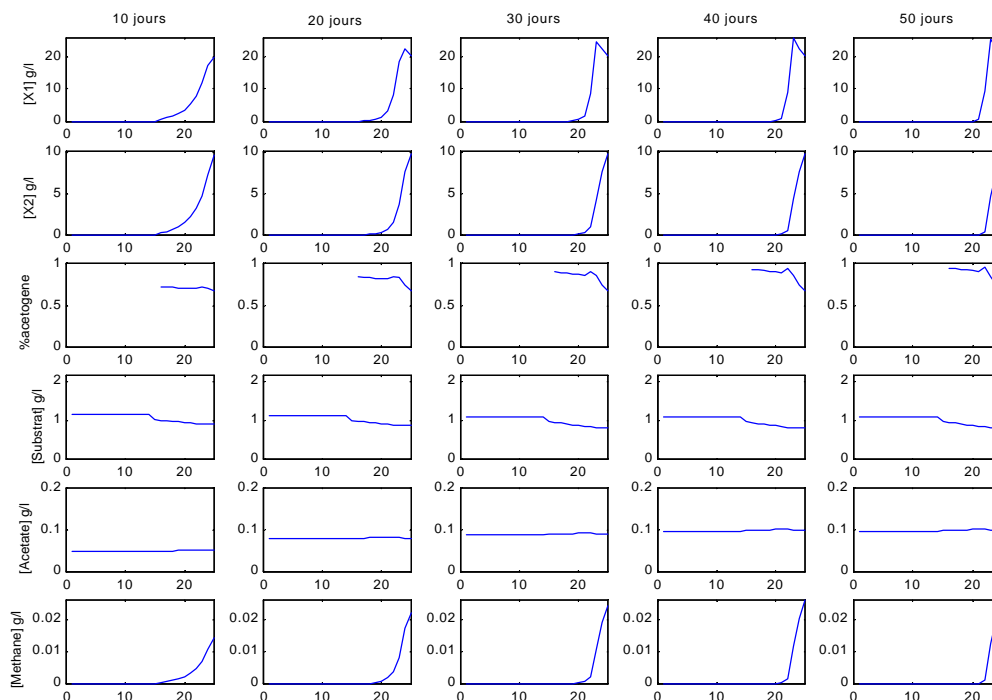


Figure 14: Profils de concentrations Simulation 2

b) Profils de concentration des biomasses

Comme le montrent la Figure 13 et la Figure 14, les profils de biomasse s'installent très vite (il est plat à l'initialisation). Le rapport des biomasses évolue selon la position au sein du biofilm, les acétogènes sont majoritaires à la périphérie du biofilm mais leur proportion diminue dans les couches profondes du biofilm. Cette répartition des biomasses est conforme aux constatations expérimentales. Dans le cas d'une alimentation riche en acides gras et pauvre en substrat, le profil de répartition des biomasses peut s'inverser, les bactéries méthanogènes ayant alors une activité plus forte que les acétogènes. Ceci est vrai tant que les acétogènes ne rentrent pas dans leur domaine d'inhibition.

c) Profils d'activité

La Figure 15 et la Figure 16 montrent l'évolution des profils d'activité du biofilm ; la première série de graphes montre l'évolution de la croissance nette, la deuxième série montre celle du taux spécifique.

La croissance nette du biofilm diminue dans le temps, ce qui laisse supposer qu'au bout des quinze jours de simulation, le système est proche de ses conditions d'équilibre. Les taux spécifiques tendent à converger vers la même valeur, ce qui signifie que le système est en équilibre, les deux biomasses atteignant un même niveau d'activité. Les taux spécifiques sont pratiquement constants au sein du biofilm, ce qui est dû aux profils plats des concentrations de solutés.

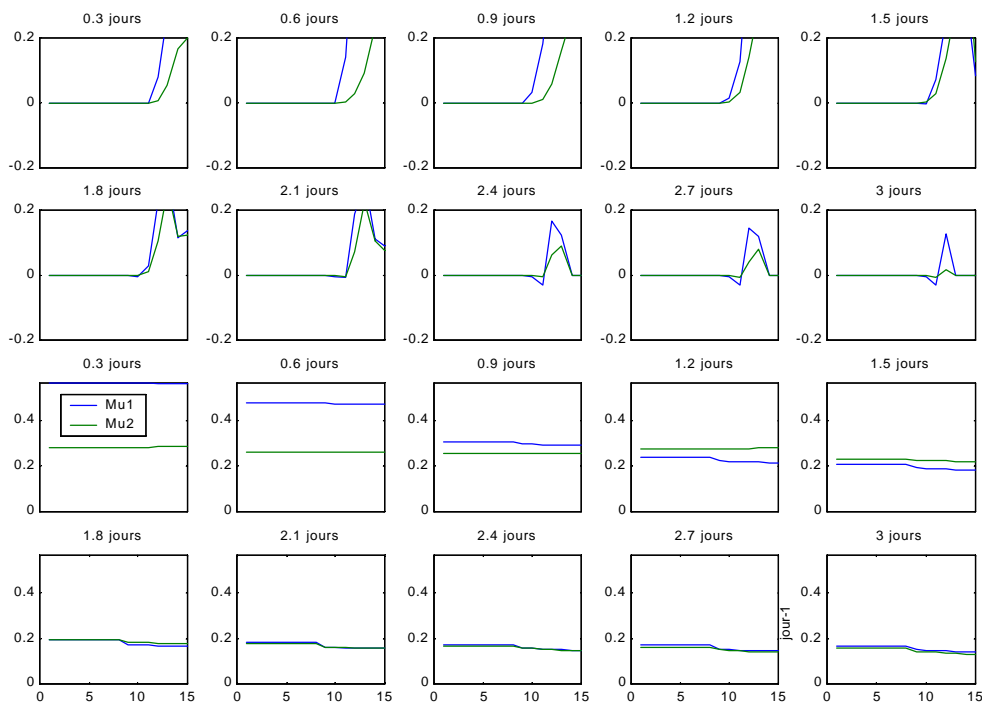


Figure 15 : Profils d'activité au sein du biofilm Simulation 1

Dans les cas où le biofilm entre en contrôle diffusif, les profils des taux spécifiques sont opposés, le taux acétogène diminue dans le biofilm alors que le taux acétogène augmente. Le profil du rapport acétogène/méthanogène peut alors être stable.

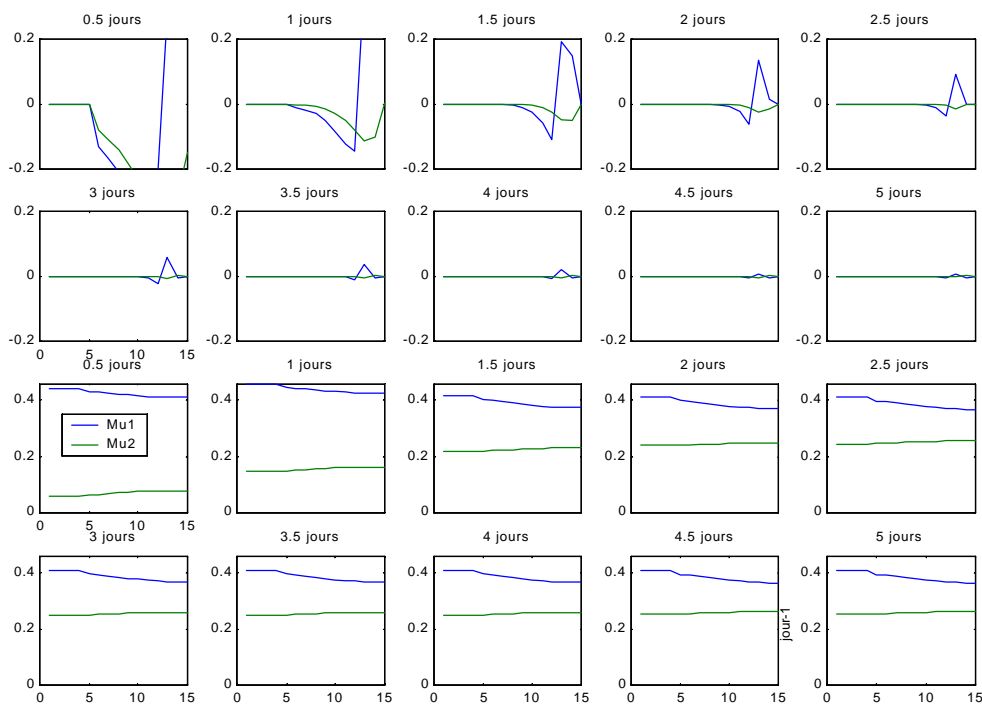


Figure 16 : Profils d'activité au sein du biofilm Simulation 2

d) Evolution du biofilm

La Figure 17 et la Figure 18 montrent l'évolution de paramètres d'état du réacteur (concentration en substrat, acétate, biomasses, Temps de séjour et débit de gaz) exprimés, pour les concentrations, en fonction du volume du biofilm et en fonction du volume du réacteur.

La croissance du biofilm se ressent sur les courbes d'évolution des concentrations en substrat et en acétate (cf. Figure 17). Dans un premier temps, le réacteur est en lessivage, les apports par l'alimentation sont largement supérieurs aux quantités métaboliques ; ceci se traduit par une convergence des concentrations au sein du réacteur vers les valeurs de l'alimentation (10g/L pour le substrat et 2 g/L pour les acides gras). A partir d'une concentration suffisante en biomasse, les quantités apportées sont consommées et les concentrations globales convergent vers une valeur faible d'équilibre. Le réacteur a alors atteint son point de fonctionnement et il traite l'effluent à dépolluer. Le passage du lessivage vers le fonctionnement normal, comparable au passage du biofilm en formation à un biofilm à l'équilibre, est visible sous la forme d'un point d'inflexion sur les courbes de concentrations globales en biomasse.

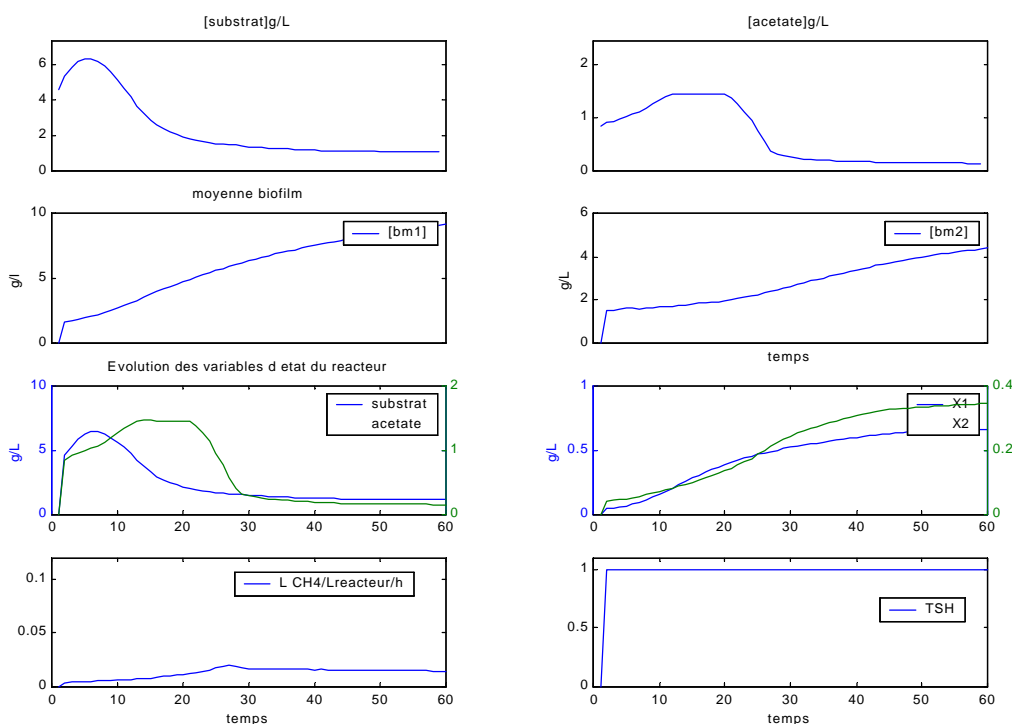


Figure 17 : Evolution des paramètres d'état vus du biofilm et vus du « réacteur » simulation 1

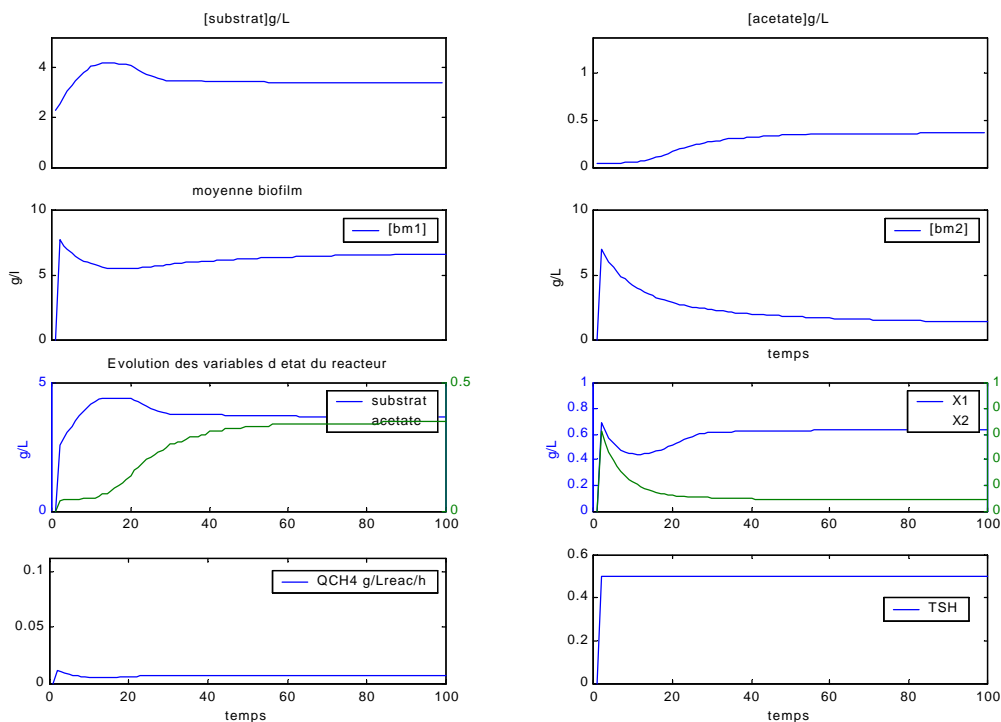


Figure 18 : Evolution des paramètres vus du biofilm et vus du réacteur simulation 2

Les courbes obtenues à partir des simulations fournissent des résultats “crédibles” et interprétables. Le modèle n’en est pas validé pour autant, puisque seules ses tendances sont validées. L’étude de sensibilité permettra de détecter d’éventuels biais du modèle.

B. Etude de sensibilité

1. CONDUITE

L’étude de sensibilité consiste à faire varier des paramètres du modèle afin de mettre en évidence quels paramètres influent fortement le modèle. Certains paramètres peuvent modifier fortement les résultats de simulations sans que cela corresponde à un résultat réel : on peut ainsi identifier les biais de modélisation et essayer de les supprimer.

Les paramètres soumis à l’étude de sensibilité sont : le pas de temps, le pas d’espace, la densité minimale de division, l’épaisseur maximale du biofilm, le nombre d’agents par cellule.

2. SENSIBILITE DES PARAMETRES DE MODELISATION

La simulation "témoin" est une simulation de 15 jours (soit 15 000 pas de temps) sur une grille spatiale de 5x15 cellules, conduite avec les valeurs de paramètres indiquées dans le paragraphe *constantes et paramètres* en tant que valeurs usuelles.

a) Masse minimale de division : Bseuil

Ce paramètre est la masse minimale à partir de laquelle une colonie dont le terme de croissance net est positif va chercher à coloniser son voisinage. Le paramètre entré est exprimé en g/L puis est calculé en grammes en ramenant la concentration au volume de la cellule spatiale. Le seuil massique est donc dépendant de la largeur de maille choisie pour la simulation. On effectue trois simulations de 15 jours avec une valeur de Bseuil respectivement de 1, 5 et 10 g/L (jeux de données 1,2 et 3).

Comme on pouvait s'y attendre, la valeur de Bseuil modifie la croissance en épaisseur du biofilm. On atteint des épaisseurs de 300 μm pour la simulation avec Bseuil=1 g/L, et seulement 200 μm pour les deux autres, pour des biofilms initiés à 100 μm dans tous les cas.

La densité massique de biomasse acétogène est plus élevée pour les simulations à 5 et 10 g/L (12 g/L biofilm au lieu de 8g/L), avec une masse totale de biomasse équivalente (de l'ordre de 0,6 g/L réacteur). La densité de biomasse méthanogène reste quasiment la même dans les trois cas.

Une valeur de Bseuil faible, en favorisant la colonisation des cellules par les agents, se traduit par une croissance du biofilm en épaisseur plus rapide au détriment de la croissance en densité.

b) Nombre d'agent maximal d'une cellule : Effsup

Ce paramètre a été introduit pour limiter le nombre d'agents. Sur les premières simulations, en effet, le nombre d'agents sur certaines cellules pouvait devenir très grand (200) amenant le blocage du système. Les valeurs testées sont de 5, 10 et 20 agents par cellules. Le cas 1 agent par cellule est un cas particulier du système multi-agents qui correspondrait à un automate cellulaire.

Les simulations diffèrent fortement par le nombre d'agents (de 150 à 500 agents) mais ont des surfaces de biofilms équivalentes. La simulation à 10 agents par cellule semble converger vers un profil du rapport acétogène/méthanogène inversé par rapport aux deux autres simulations. Cette simulation diffère également par l'évolution des concentrations des solutés. Notons cependant que les concentrations globales des biomasses sont comparables pour toutes les simulations (0,5 g/L acétogène et 0,2 g/L méthanogène) ; les différences observées sont peut-être des différences transitoires.

Il faudra vérifier, d'une part l'effet de simulations plus longues et l'effet d'une densité maximale en agents radicalement plus grande, au risque de déstabiliser le simulateur.

c) Epaisseur du biofilm : Thickness

La valeur Thickness détermine en fait le terme de disparition de biomasse puisqu'il indique l'épaisseur de biofilm correspondant à la perte de biomasse observée. Des valeurs élevées de Thickness déterminent des pertes de biomasse faibles. Pour les valeurs élevées on observe logiquement une croissance plus rapide du biofilm. La valeur la plus élevée (800 μm) conduit à un biofilm où la biomasse acétogène épuise le substrat à sa disposition, la biomasse méthanogène métabolise alors surtout les acides gras issus de l'alimentation du réacteur. L'activité méthanogène étant dans ce cas plus importante que l'activité acétogène, le profil du rapport des biomasses tend à s'aplanir voire à s'inverser.

d) Pas de temps : Dt

Aucune différence n'a pu être mise en évidence entre trois simulations réalisées avec des pas de temps respectifs de 1, 5 et 10 millièmes de jour (soit environ 1 minute et demi, 7 minutes et 14 minutes). L'algorithme de calcul par la méthode ADI arrive à la solution sans nécessiter de trop nombreuses itérations.

e) Pas spatial : DI

On compare cinq simulations, effectuées avec des largeurs de maille de 5,10, 20, 50 et 100 μm . A chaque fois on initie la simulation avec un biofilm homogène de 100 μm d'épaisseur. Les simulations fournissent, après 15 000 pas, des biofilms d'épaisseurs différentes (de 130 à 400 μm). Il semble donc que pour des largeurs de maille élevée, la croissance en épaisseur soit privilégiée par rapport à la croissance en densité.

Les règles de distribution de biomasse ne sont donc pas valides ; des agents sur de petites cellules ont un pouvoir colonisant inférieur à ceux présents sur une grande cellule. Ceci est peut être dû à la saturation en biomasse des cellules

3. SENSIBILITE A LA SEQUENCE DES CALCULS

La modélisation de la dynamique rapide, c'est-à-dire des phénomènes de réaction et de diffusion doit effectuer pour tout point de l'espace les calculs suivants :

- Activité spécifique des agents
- Réaction biochimique locale
- Diffusion des molécules

Au cours de l'élaboration du modèle, plusieurs méthodes de calcul ont été essayées. Dans la première version du modèle, sans passerelle avec Matlab, les trois calculs étaient effectués de manière séquentielle mais les phénomènes restaient synchrones. Avec la mise en place de la passerelle (BIOFILM v2) les trois phases du calcul sont conduites en une seule étape. La considération de l'activité métabolique propre de chaque agent (BIOFILM v3) oblige, pour chaque calcul d'activité, à interroger chaque agent. En délocalisant le calcul de la réaction diffusion vers Matlab, il est impossible d'accéder aux variables de chaque agent. L'activité moyenne de chaque cellule est alors calculée depuis la dynamique lente, en faisant l'hypothèse que la valeur d'activité n'évoluera pas de manière significative pendant un pas de temps de dynamique lente. Cette hypothèse est raisonnable car les courbes de l'activité spécifique évoluent assez lentement en fonction des concentrations. Cependant cette hypothèse est à vérifier dans chaque nouvelle conformation.

Le cycle de calcul de dynamique rapide est effectué plusieurs fois au sein d'un cycle de dynamique lente. Au cours de l'élaboration du modèle, plusieurs séquences de calcul ont été testées :

Dynamique			
lente		Rapide	
Croissance		Activité	Réaction Diffusion
Croissance		Activité-Réaction-Diffusion	
Croissance	Activité	Réaction-Diffusion	
Croissance	Activité	Réaction-Diffusion optimisé	

La dernière méthode est la plus rapide en terme de temps de calcul.

Les comparaisons entre des algorithmes calculant la réaction et la diffusion en une seule résolution (v2) ou de manière séquentielle (v3) ont montré peu de différences, à différentes largeurs de maille comme l'indique le tableau. Tout au plus le champ de concentration est légèrement moins lisse dans le cas de la résolution séquentielle.

Maille	Cycles		Différence S1	Différence S2
	D-R	D/R		
50 μm			0,15% - 0,8%	0,11% - 0,33%
5 μm	175	161	0,64% - 2,9%	0,28% - 0,91%

Tests : calcul pour 0.001 jour sur une matrice 30*30

C. Développements ultérieurs du modèle

1. DEVELOPPEMENTS DU MODELE

Ce premier modèle a laissé de côté de nombreux phénomènes, visant la validation d'une approche méthodologique. Ses limites montrées par l'analyse de sensibilité ainsi que sa confrontation aux connaissances actuelles des biofilms nous pousse à ajouter au modèle actuel d'autres mécanismes et d'autres phénomènes.

a) Un modèle plus proche des connaissances biologiques

Par rapport au niveau de connaissances actuel sur le fonctionnement des biofilms, nous avons écarté beaucoup de phénomènes. Parmi les phénomènes majeurs qu'il faudrait ajouter, il y a la production de matrice extracellulaire, la dormance d'une part de la biomasse et les communications intercellulaires.

Le mécanisme de disparition de biomasse tel qu'il est décrit actuellement ne se base pas sur la simulation de phénomènes microscopiques pour reproduire un phénomène macroscopique mais sur l'application d'une règle macroscopique. Une modélisation du détachement cellulaire basée sur l'adhésion à la matrice, la mort cellulaire et les forces de détachement serait plus représentative des phénomènes à l'origine de la disparition de biomasse. Il paraît de toute manière nécessaire de modéliser un composant qui représente jusqu'à 70% de la masse d'un biofilm. Une première amélioration du modèle consisterait donc à lui ajouter une nouvelle composante « EPS », probablement distribuée entre les propriétés de l'espace et celles des agents.

Les phénomènes d'inactivation sont très importants dans la dynamique des biofilms. C'est en effet l'existence de bactéries dormantes, capables de reprendre leur activité dès que nécessaire, qui permet aux biofilms d'être si résistants aux perturbations extérieures.

La prise en compte de ces phénomènes permettrait d'aboutir à un modèle beaucoup plus représentatif d'un biofilm mais nécessite l'identification de nombreux paramètres. Leur implémentation nécessite donc la conduite d'expériences spécifiques en parallèle.

b) De nouvelles règles pour la distribution de biomasse

Les règles de distribution de biomasse utilisées ont été invalidées par l'analyse de sensibilité. Il faut donc les repenser. Le modèle établi par Kreft reposait sur le déplacement des bactéries

selon des règles de minimisation du recouvrement des bactéries entre elles, selon une technique similaire aux Boids (simulation multi-agents des troupeaux) de Reynolds. Dans le modèle BIOFILM, en considération de la taille des agents et des forces d'adhésion de la matrice extracellulaire, la biomasse était considérée comme fixée. Manifestement cette hypothèse doit être revue car elle introduit un fort ralentissement de la croissance en épaisseur pour les mailles fines : en effet, actuellement seule les agents situés à l'interface peuvent participer à la croissance en épaisseur. Le transfert de biomasses aux colonies filles devait réduire ce phénomène mais manifestement il n'a pas suffi.

Un modèle validé basé sur un automate cellulaire [Picioreanu *et al* 97] déplace la biomasse de cellule en cellule. Lorsque tous les voisins d'une cellule sont saturés, une des cellules voisines est poussée pour libérer de l'espace.

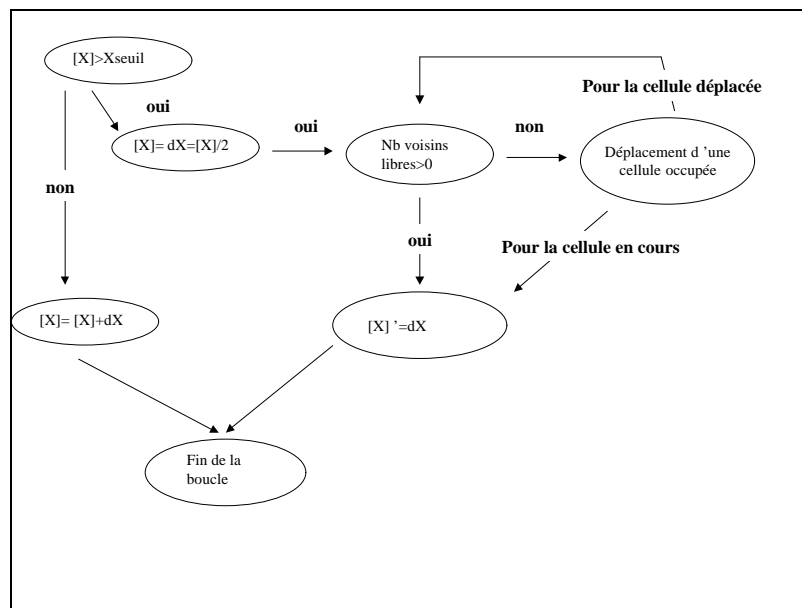


Figure 19 : Règles de distribution de biomasse appliquées par Picioreanu

Le modèle de Wik [Wik 97], réalisé sur un filtre anaérobie, inclut lui aussi un déplacement des biomasses.

Les nouvelles règles de distribution des biomasses devront sans doute inclure ces phénomènes.

c) Espèces chimiques

Actuellement, peu d'espèces chimiques sont suivies par le modèle et seules les réactions biochimiques sont simulées alors que les équilibres acido-basiques et gazeux jouent un rôle non négligeable dans l'activité du réacteur. D'autre part la modélisation de la dynamique de ces substances permettrait de produire plus d'indicateurs comparables à ceux utilisés pour le suivi d'un réacteur (alcalinité, %CO₂, pH,...) et donc d'affiner l'interprétation en terme de dynamique du biofilm des mesures actuellement obtenues.

d) Algorithmes génétiques

Deux phénomènes observés sur les biofilms poussent à s'intéresser à l'application de la théorie des algorithmes génétiques aux agents bactériens :

- D'une part, les populations bactériennes peuplant le biofilm subissent une sélection en fonction des contraintes appliquées. Lors de la mise en place du biofilm, l'application de contraintes fortes sélectionne les populations les plus actives.
- D'autre part, les voies métaboliques actives évoluent selon l'état du biofilm : pendant la mise en place du biofilm, le substrat dégradé sert surtout à la production de matrice extracellulaire alors que lorsque le biofilm est stable, le substrat est dégradé majoritairement en méthane.

Ces phénomènes pourraient être modélisés par l'écriture des paramètres métaboliques sur deux « gènes », qui seraient activés selon l'état du biofilm. Ces caractères métaboliques héréditaires permettraient de simuler la concurrence entre les agents et la sélection des plus performants.

2. LIMITES DE LA PLATE-FORME UTILISEE

La plate-forme CORMAS dans son architecture, ne présente pas d'incompatibilité avec le type de modèle proposé. L'architecture globale du modèle BIOFILM pour CORMAS rejoint d'ailleurs celle de BACSIM, le modèle réalisé par Kreft et Booth.

Cependant l'implémentation du modèle a soulevé quelques problèmes : le recours à un logiciel spécialisé (MATLAB) pour effectuer une part des calculs s'est avérée accélérer notablement les calculs mais au prix d'une instabilité chronique du logiciel. Cette instabilité serait peut être levée par l'utilisation d'un autre type de passerelle entre CORMAS et Matlab que les liens DDE.

Enfin, il est nécessaire de rappeler, même si cela semble inévitable pour ce genre de travaux, que les temps de calcul d'une simulation sont longs ; il faut en effet compter 2h30 de calcul pour une grille de 750 sur 250 microns, avec une maille de 50 μm alors que pour des simulations plus réalistes, il faudrait affiner la largeur des mailles à 5 μm en maintenant la taille de l'espace modélisé.

VI. Conclusion

Le but de ce travail était la réalisation d'un modèle de biofilm appliqué au cas des réacteurs anaérobies. Les biofilms interviennent dans la digestion anaérobie mais aussi dans de très nombreux domaines industriels et médicaux.

La digestion anaérobie repose sur l'activité de deux groupes de micro-organismes, les bactéries acétogènes et les archéobactéries méthanogènes. Ces deux groupes sont organisés en chaîne trophique, les méthanogènes utilisant les acides gras produits par les acétogènes. Parmi les différentes architectures de réacteur, beaucoup utilisent la propriété des micro-organismes à se fixer sur les parois sous forme de biofilms. Ces biofilms sont des structures complexes, assemblages de bactéries et de matrice extracellulaire, caractérisées par une forte hétérogénéité et une stratification.

La connaissance et la maîtrise de l'écosystème que forment les biofilms est une étape importante pour la conduite optimale des réacteurs mais aussi pour la lutte contre la formation des biofilms indésirables.

La modélisation permet de fournir des prévisions sur l'évolution du système et ses réactions aux changements du milieu. Le type de modélisation choisi est de plus en plus utilisé pour les modélisations écologiques. La modélisation multi-agent se base sur l'hypothèse que les propriétés d'un système sont déterminées par les comportements et les interactions qui le composent. En s'appuyant sur les mécanismes au niveau de l'individu, elle permet de rendre compte de l'hétérogénéité du système et de traduire directement les connaissances sur les comportements des individus. A l'approche classique de modélisation, elle apporte l'avantage de mettre en évidence le niveau de connaissances du système et de faire apparaître les phénomènes insuffisamment connus.

Conformément à l'archétype d'un système multi-agents, le modèle BIOFILM définit un environnement sur lequel évoluent des agents. Les agents sont définis comme des entités réactantes, assimilables à une colonie homogène de bactéries. Cette entité n'est pas forcément observée mais elle doit identifier le comportement d'un ensemble de bactéries tout en gardant une dimension d'agent. Le recours à un agent virtuel permet de ne pas manipuler un nombre trop important d'agents. Chaque agent est fixé, suit une balance massique et est capable de créer de nouveaux agents dans son voisinage.

La dynamique de l'environnement est celle du champ de concentrations des différentes espèces chimiques. La dynamique du champ de concentration est modélisée par un phénomène de diffusion-réaction au sein du biofilm, et par une zone parfaitement mélangée dans la phase libre. La résolution du système différentiel de réaction diffusion est effectuée en sur un plan, selon la méthode des directions alternées implicites, qui converge rapidement vers la solution.

L'étude de sensibilité a montré que le choix de la largeur des mailles influençait fortement les résultats. Ce biais de simulation est imputé aux règles de distribution de biomasse et de création des nouveaux agents. De nouvelles règles devront être établies, basée sur l'hypothèse que la biomasse n'est pas fixée mais susceptible de se déplacer sous la pression des voisins.

La prise en compte de la matrice extracellulaire devrait également permettre d'affiner les règles décrivant l'arrachement de biomasse ainsi que le champ de diffusivité. Une amélioration du modèle passera aussi par l'augmentation du nombre d'espèces chimiques modélisées et par la dotation de plusieurs voies métaboliques possibles aux agents bactériens.

Ce travail présente l'architecture d'un modèle multi-agents de biofilm. Si le modèle n'est pas encore validé et demande notamment une refonte des règles de distribution de biomasse, l'approche est prometteuse car capable de représenter l'hétérogénéité d'un biofilm et de se baser sur ses mécanismes fondamentaux.

VII. Bibliographie

- Bernard O., Hadj-Sadok Z., Dochain D., Genovesi A., Steyer J-P.** (2001) Dynamical model development and parameter identification for an anaerobic wastewater treatment process. *Biotechnology Engineering in press*
- Droste R. L., Kennedy K.J.** (1986) Sequential Substrate utilization and effectiveness factor in fixed biofilms, *Biotechnology and Bioengineering* vol XXVIII, pp 1713-1720
- Kreft J-U, Booth G., Wimpenny J.W.T.** (1998) BacSim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, 144, 3275-3287
- Kreft J-U, Booth G., Wimpenny J.W.T.** (1999) Application of individual-based modelling in microbial ecology, in *Proceedings of the 8th International Symposium on Microbial Ecology*
- Kreft J-U, Booth G., Picioreanu C.** (2001) Individual-based modelling of biofilms, *Microbiology*, in press
- Le Page C., Bousquet F., Bommel P., Baron C., Lardon S.** (2001) CORMAS : un environnement de développement de systèmes multi-agents dédié à la gestion de ressources naturelles, *Techniques et Sciences informatiques*, Volume X, n°X/2001, 1-19
- Picioreanu C., van Loosdrecht M.C.M., Heijnen J.J.** (1998a) A new combined differential discrete cellular automaton approach for biofilm modeling : application for growth in gel beads. *Biotechnology and Bioengineering* 57(6), 719-731.
- Picioreanu C., van Loosdrecht M.C.M., Heijnen J.J.** (1998b) Mathematical Modeling of Biofilm structure with a hybrid differential discrete cellular automaton approach *Biotechnology and Bioengineering* 58(1), 101-116.
- Wik T.** (1999) *On modelling the dynamics of fixed biofilm reactors*, Ph.D. thesis, Chalmers University of technology, Goteborg
- Wimpenny J., Manz W., Szewzyk U.** (2000) Heterogeneity in biofilms *FEMS Microbiology Reviews* 24, 661-671.
- Zhang T., Bishop P.** (1994a) Density, porosity and pore structure of biofilms. *Water Research* 28, 2267-2277

VIII. Annexes

1. Principe de l'ADI
2. Code source de l'algorithme de résolution par la méthode ADI (Matalb 5)
3. Code source du modèle BIOFILM (CORMAS)

ADI

La méthode des Directions Implicites Alternées a été proposée par **Peaceman et Rachford** en 1955 pour la résolution des problèmes de diffusion réaction en plusieurs dimensions. Cette méthode a l'avantage de ne pas reposer sur une hypothèse de stabilité.

La relation différentielle décrivant la diffusion-réaction s'écrit :

$$\left. \frac{\partial S}{\partial t} \right|_{x,y} = \left(D \frac{\partial^2 S}{(\partial X)^2} + r(X, S) \right) dt$$

où ρ désigne la variation de concentration due à l'activité biologique (classiquement $\rho=k\mu X$).

Une approximation par les différences finies de cette relation s'écrit :

$$\left\{ \begin{array}{l} d_t S_{x,y} = \left(\frac{D}{dt^2} \left(\frac{d_x^2 S}{(dX)^2} + \frac{d_y^2 S}{(dY)^2} \right) + r(X_{x,y}^t, S_{x,y}^t) \right) dt \\ d_t S_{x,y} = S_{x,y}^{t+dt} - S_{x,y}^t \\ d_x^2 S = S_{x+1,y} - 2S_{x,y} + S_{x-1,y} \end{array} \right.$$

Le théorème de l'ADI permet d'exprimer cette équation aux différences finies sous forme d'un système de deux différences finies appliquées dans deux directions différentes.

$$\left\{ \begin{array}{l} S_{x,y}^{t+dt/2} - S_{x,y}^t = D \frac{dt}{2} \left(d_x^2 S_{x,y}^{t+dt/2} + d_y^2 S_{x,y}^t + r(X_{x,y}^t, S_{x,y}^t) \right) \\ S_{x,y}^{t+dt} - S_{x,y}^{t+dt/2} = D \frac{dt}{2} \left(d_x^2 S_{x,y}^{t+dt/2} + d_y^2 S_{x,y}^{t+dt} + r(X_{x,y}^t, S_{x,y}^{t+dt/2}) \right) \end{array} \right.$$

En factorisant la première équation, on obtient la relation :

$$\begin{aligned} (-j) S_{x-1,y}^{t+dt/2} + (1+2j) S_{x,y}^{t+dt/2} + (-j) S_{x+1,y}^{t+dt/2} &= (j) S_{x,y-1}^t + (1+2j) S_{x,y}^t + (j) S_{x,y+1}^t + r(X_{x,y}^t, S_{x,y}^t) \\ j &= D \frac{dt}{2d^2} \end{aligned}$$

Appliquée à toutes les cellules d'une ligne (c'est à dire toutes les valeurs de x et une valeur de y) on obtient un système d'équations dont la solution est une ligne du plan au temps $t+\delta t/2$. En balayant le plan selon l'axe y, on obtient autant de systèmes d'équations qui permettent de décrire le champs de concentrations au temps $t+\delta t/2$.

De même, la deuxième équation permet de générer une série de systèmes en balayant le plan selon l'axe x.

$$(-j) S_{x,y-1}^{t+dt} + (1+2j) S_{x,y}^{t+dt} + (-j) S_{x,y+1}^{t+dt} = (j) S_{x-1,y}^{t+dt/2} + (1+2j) S_{x,y}^{t+dt/2} + (j) S_{x+1,y}^{t+dt/2} + r(X_{x,y}^t, S_{x,y}^{t+dt/2})$$

Cependant, sous cette forme les systèmes d'équations n'ont pas de solution unique : chaque ligne comprenant 3 inconnues « alignées », on ne possède pour n inconnues que $n-2$ relations, les inconnues situées en bordure du plan ne possédant qu'une voisine sur un axe. Il faut donc ajouter des conditions aux bordures afin de lever l'indétermination. Nous avons appliqué pour l'axe x (normal au biofilm) les hypothèses classiques sur les conditions aux bordures utilisées en modélisation des biofilms citées plus haut, et pour l'axe y un bouclage de l'environnement selon cette direction. Cette dernière condition repose sur l'hypothèse qu'il n'y a pas de gradient selon l'axe y ; cette hypothèse est valide pour des simulations de petite surface.

Les systèmes d'équations sont triangulaires, on a donc intérêt à effectuer une résolution gaussienne par élimination plutôt qu'une résolution par inversion de matrice. Ceci se traduit par l'utilisation de l'opérateur « \backslash » plutôt que « \wedge^{-1} » dans Matlab et permet un important gain de temps de calcul surtout pour les matrices de grande taille.

```

%***** diffusionR *****
% Laurent Lardon 24/08/2001
% INRA LBE Narbonne
%*****
% Modélise la dynamique rapide d'un biofilm
% Phénomènes de réaction-diffusion en une passe avec pas de temps adaptable
%*****

function reponse=diffusionR(matrice,L,N,dt,S0,Ac0)
%close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Chargement des données et des paramètres $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

S1=flip(L,N,matrice(:,1));
S2=flip(L,N,matrice(:,2));
S3=zeros(L,N);
X1=flip(L,N,matrice(:,3));
X2=flip(L,N,matrice(:,4));

load('param.txt','-ascii');

% Parametres du reacteur
Sin=param(2);
Acin=param(4);
TS=param(5); % temps de séjour hydraulique en h
ASpe=param(6); % surface spécifique du reacteur

% Diffusivités
D1=param(7); %0.6e-4; %m²/j glucose
D2=param(8); %1.03e-4; %m²/j acétate

%constantes de modelisation
dl=param(9); %5e-6; %largeur maille en m
%paramètres métaboliques
Ks1=param(10); k11=param(11);k12=param(12);
Ks2=param(13);k21=param(14);k22=param(15);
Ki2=param(16);
mu1=param(17);mu2=param(18);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Préparation des données $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%passage en adimensionnel
S1=S1/Sin;
S2=S2/Sin;

%calcul du tau basé sur tau pour la diffusion d'une grille aléatoire;
tau=dl^2*4e4; %pas de temps de la boucle de calcul

%initialisation des compteurs
temps=0;
index=1;

%boucles
while temps<dt

%***** Application ADI *****

grille1=S1;grilleX1=X1;
grille2=S2;grilleX2=X2;
x=size(grille1,2); y=size(grille1,1);
buffS1=zeros(y,x);buffS2=zeros(y,x);

%preparation du 1er systeme d'eqn
%S1
D=D1;

%1er membre
jean=member1(D1,tau,x,dl);

```

```

% ecriture du deuxieme membre et résolution des systemes pour tout y
% bouclage de l'environnement le long de l'axe y
phi=zeros(x,1);
phi(1,1)=S0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille1(y,2:x-1)+(1-D*tau/(dI^2))*grille1(1,2:x-1)+ ...
(D*tau/(2*dI^2))*grille1(2,2:x-1)+...
(k11*grilleX1(1,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS1(1,:)=(jean\phi);

for j=2:y-1
phi=zeros(x,1);
phi(1,1)=S0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille1(j-1,2:x-1)+(1-D*tau/(dI^2))*grille1(j,2:x-1)+ ...
(D*tau/(2*dI^2))*grille1(j+1,2:x-1)+...
(k11*grilleX1(j,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS1(j,:)=(jean\phi);
end

phi=zeros(x,1);
phi(1,1)=S0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille1(1,2:x-1)+(D*tau/(2*dI^2))*grille1(y-1,2:x-1)+...
(1-D*tau/(dI^2))*grille1(y,2:x-1)+...
(k11*grilleX1(y,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS1(y,:)=(jean\phi);

%figure;surf(buffS1);

%cas S2
D=D2;
%preparation du 1er systeme d'eqn
%1er membre
jean=member1(D2,tau,x,dI);

% ecriture du deuxieme membre et résolution des systemes pour tout y
% bouclage de l'environnement le long de l'axe y
phi=zeros(x,1);
phi(1,1)=Ac0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille2(y,2:x-1)+(1-D*tau/(dI^2))*grille2(1,2:x-1)+ ...
(D*tau/(2*dI^2))*grille2(2,2:x-1)+...
(k12*grilleX1(1,2:x-1)*tau/2/Sin)+...
(k21*grilleX2(1,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS2(1,:)=(jean\phi);

for j=2:y-1
phi=zeros(x,1);
phi(1,1)=Ac0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille2(j-1,2:x-1)+(1-D*tau/(dI^2))*grille2(j,2:x-1)+ ...
(D*tau/(2*dI^2))*grille2(j+1,2:x-1)+...
(k12*grilleX1(j,2:x-1)*tau/2/Sin)+...
(k21*grilleX2(j,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS2(j,:)=(jean\phi);
end

phi=zeros(x,1);
phi(1,1)=Ac0/Sin;
phi(2:x-1,1)=(D*tau/(2*dI^2))*grille2(1,2:x-1)+(D*tau/(2*dI^2))*grille2(y-1,2:x-1)+...
(1-D*tau/(dI^2))*grille2(y,2:x-1)+...
(k12*grilleX1(y,2:x-1)*tau/2/Sin)+...
(k21*grilleX2(y,2:x-1)*tau/2/Sin);
phi(x,1)=0;
buffS2(y,:)=(jean\phi);

%passage a la transposée
buffS1=buffS1';
grilleX1=grilleX1';
buffS2=buffS2';
grilleX2=grilleX2';

%preparation du 2e systeme d'eqn
%cas S1

```

```

D=D1;
jean=member2(D1,tau,y,dl);

% ecriture et résolution des systemes pour tout x
phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*ones(y,1)*S0/Sin +(1-D*tau/(dl^2))*buffS1(1,:)'+(D*tau/(2*d1^2))*buffS1(2,:)' +...
(k11*grilleX1(1,:)*tau/2/Sin)';
grille1(:,1)=(jean\phi);
grille1(:,1)=S0/Sin;

for j=2:x-1
phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*buffS1(j-1,:)'+(1-D*tau/(dl^2))*buffS1(j,:)'+(D*tau/(2*d1^2))*buffS1(j+1,:)' +...
(k11*grilleX1(1,:)*tau/2/Sin)';
grille1(:,j)=(jean\phi);
end

phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*buffS1(x-1,:)'+(1-D*tau/(dl^2))*buffS1(x,:)'+(D*tau/(2*d1^2))*buffS1(x-1,:)' +...
(k11*grilleX1(1,:)*tau/2/Sin)';
grille1(:,x)=(jean\phi);
grille1(:,x)=grille1(:,x-1);
%figure;surf(grille1);

%cas S2
D=D2;
jean=member2(D2,tau,y,dl);
% ecriture et résolution des systemes pour tout x
phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*ones(y,1)*Ac0/Sin +(1-D*tau/(dl^2))*buffS2(1,:)' +...
(D*tau/(2*d1^2))*buffS2(2,:)' +...
(k12*grilleX1(1,:)*tau/2/Sin)'+...
(k21*grilleX2(1,:)*tau/2/Sin)';
grille2(:,1)=(jean\phi);
grille2(:,1)=Ac0/Sin;

for j=2:x-1
phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*buffS2(j-1,:)'+(1-D*tau/(dl^2))*buffS2(j,:)' +...
(D*tau/(2*d1^2))*buffS2(j+1,:)' +...
(k12*grilleX1(1,:)*tau/2/Sin)'+...
(k21*grilleX2(j,:)*tau/2/Sin)';
grille2(:,j)=(jean\phi);
end

phi=zeros(y,1);
phi=(D*tau/(2*d1^2))*buffS2(x-1,:)'+(1-D*tau/(dl^2))*buffS2(x,:)' +...
(D*tau/(2*d1^2))*buffS2(x-1,:)' +...
(k11*grilleX1(x,:)*tau/2/Sin)'+...
(k21*grilleX2(x,:)*tau/2/Sin)';
grille2(:,x)=(jean\phi);
grille2(:,x)=grille2(:,x-1);
%redressement de la grille
grilleX1=grilleX1';
grilleX2=grilleX2';

%Indicateurs
indic(index,1)=sum(sum(abs(grille1-S1)));
indic(index,2)=sum(sum(abs(grille2-S2)));

S1=grille1;S2=grille2;
temps=temps+tau;

%Optimisation du pas de temps
if max(indic(index,1),indic(index,2))<0.2
tau=tau*5;
end
tau=min(tau, dt/10);
tau=min(tau, dt-temps);
index=index+1;

end

%repassage en dimensionnel
S1=S1*Sin;

```

```
S2=S2*Sin;
S3=k22*grilleX2*dt;
```

```
%repassage en matrice colonnes
reponse=flop(L,N,S1);
reponse(:,2)=flop(L,N,S2);
reponse(:,3)=flop(L,N,S3);
```

```
%***** Subfunctions *****
```

```
function m=member1(D,tau,x,dl)
a1=-(D*tau/(2*dl^2));
a2=1+D*tau/(dl^2);
a3=a1;
d1=diag(ones(1,x-1),-1);
d2=diag(ones(1,x),0);
d3=diag(ones(1,x-1),1);
m=a1*d1+a2*d2+a3*d3;
m(1,1)=1;m(1,2)=0;m(1,3)=0;
m(x,x)=-1;m(x,x-1)=1;
```

```
function m=member2(D,tau,y,dl)
a1=-(D*tau/(2*dl^2));
a2=1+D*tau/(dl^2);
a3=a1;
d1=diag(ones(1,y-1),-1);
d2=diag(ones(1,y),0);
d3=diag(ones(1,y-1),1);
m=a1*d1+a2*d2+a3*d3;
m(1,y)=a3;
m(y,1)=a1;
```

```
function reponse=flip(L,N,mat)
%molinette sur les matrices : passe en matrice spatiale
for i=1:L*N
    j=fix((i-1)/N)+1;      %n° de ligne
    k=i-((j-1)*N);        %n° de colonne
    reponse(j,k)=mat(i);
    %reponse(j,k+2*N)=mat(i,3);
    %reponse(j,k+3*N)=mat(i,4);
end
```

```
function reponse=flop(L,N,mat)
%molinette sur les matrices : passe en matrice tabulaire
for i=1:L*N
    j=fix((i-1)/N)+1;      %n° de ligne
    k=i-((j-1)*N);        %n° de colonne
    reponse(i,1)=mat(j,k);
end
```

SpatialEntity_Element subclass: #Biofilm_Cell

```
instanceVariableNames: 's1 s2 s3 xi1 xi2 film buffs1 buffs2 buffs3 bm bm1 bm2 effectif '
classVariableNames: "
poolDictionaries: "
category: 'Cormas-Biofilm'
```

!Biofilm_Cell methodsFor: 'controle'!**step:repet**

```
"Stocke dans les buffers les influences de chacun des processus qui feront la valeur du temps t+1"
self diffusion:repet.
self reaction:repet.
"self charge."!!
```

!Biofilm_Cell methodsFor: 'procedure'!**bilanBM**

```
"normalise les concentrations"
| t |
t:=0.
self theOccupants do: [:membre| membre do: [:colonie| t:=t+(colonie m/(DI*DI*Df*0.001))].
self theOccupants do: [:membre| membre do: [:colonie| (t>(BMax)) ifTrue:[colonie m: colonie m/t*BMax]]].
```

"calcul de la biomasse totale et du nombre de colonies de la cellule"

```
self bm1:0.
self bm2:0.
self effectif: 0.
self xi1: 0.
self xi2: 0.
(self theOccupants at: #Biofilm_Aceto) do: [:colonie|
    self bm1: self bm1+colonie m.
    self effectif: self effectif +1.
    self xi1: self xi1+(colonie mu*colonie m)].
(self theOccupants at: #Biofilm_Methano) do: [:colonie|
    self bm2: self bm2+colonie m.
    self effectif: self effectif +1.
    self xi2: self xi2+(colonie mu*colonie m)].

self bm1: (self bm1/(DI*DI*Df*0.001)).
self xi1: (self xi1/(DI*DI*Df*0.001)).
self bm2: (self bm2/(DI*DI*Df*0.001)).
self xi2: (self xi2/(DI*DI*Df*0.001)).
self bm: self bm1+self bm2.!
```

bilanFilm

```
"Met a jour la variable film: true pour le biofilm et la couche limite"
[cellFilm]
cellFilm:=0.
```

```
self neighbourhood do:[:cellule|(cellule bm1 + cellule bm2)>0 ifTrue:[cellFilm:=cellFilm+1]].
```

```
(cellFilm>0)    ifTrue:[self film:true]
                ifFalse:[self film:false].
(self bm>0) ifTrue: [self film:true].!
```

charge:repet

```
"alimentation-dilution des cellules appartenant à la phase liquide du reacteur"
(self bm=0) ifTrue: [
    self buffs1: self buffs1+((Sin-self s1)/TSH*Dt/repet).
    self buffs2: self buffs2+((0-self s2)/TSH*Dt/repet).].!
```

diffusion:repet

```
"diffusion des molécules selon loi de Fick discretise"
```

```
| voisin total diff |
voisin := self neighbourhood.
```

```
"diffusion acetate"
```

```
"selection du taux de diffusion en fonction de l'appartenance au biofilm"
self film ifTrue: [diff := (D1Ac*(-0.5*(self bm min:BMax)/BMax+0.8)].
self film ifFalse: [diff := D2Ac].
```

```
"calcul de la qte de matiere mobilisee par le processus de diffusion sur la cellule"
total := 0 asDouble.
```



```

voisin do: [:cellule | total := total + (cellule s2 - self s2)].
((total abs)<(1d-100)) ifTrue: [total:=0].
self buffs2: self buffs2 + (diff * Dt * total/ (repet*Dl * Dl) ).

```

```

"diffusion substrat"
"selection du taux de diffusion en fonction de l'appartenance au biofilm"
self film ifTrue: [diff := (D1DCO*(-0.5*(self bm1+self bm2 min:BMax)/BMax+0.8))].
self film ifFalse: [diff := D2DCO].
"calcul de la qte de matiere mobilisee par le processus de diffusion sur la cellule"
total:=0.
voisin do: [:cellule | total := total + (cellule s1 - self s1)].
self buffs1: self buffs1 + (diff * Dt * total / (repet*Dl * Dl)).!

```

influence

"met a jour les variables avec leur buffer"

```

self s1: (self s1 + self buffs1) asDouble.
self s2:(self s2 +self buffs2) asDouble.
self s3: self buffs3."

```

"Remise a zero des buffers"

```

self buffs1: 0 asDouble.
self buffs2: 0 asDouble.
self buffs3: 0 asDouble.!

```

reaction: repet

"Reaction bacterienne fast dynamic"

```

"Calcul du mu de la cellule en fonction du mu de chaque colonie"
(self theOccupants at: #Biofilm_Aceto) do: [:colonie |
    colonie calculMu.
    colonie reaction:repet.].

```

```

(self theOccupants at: #Biofilm_Methano) do: [:colonie |
    colonie calculMu.
    colonie reaction:repet.].! !

```

!Biofilm_Cell methodsFor: 'init'!

init

```

self initliquide.!

```

initliquide

"initialise les cellules de l'environnement de la phase liquide"

```

super init.
self s1: S0 asDouble.
self s2: (Ac0) asDouble.
self s3: 0 asDouble.
self film: false.
self buffs1: 0 asDouble.
self buffs2: 0 asDouble.
self buffs3: 0 asDouble.

```

```

self bm1:0.
self bm2:0.
self bm:0!!

```

!Biofilm_Cell methodsFor: 'accessing'!

```

bm
    ^bm!

```

```

bm1
    ^bm1!

```

```

bm1: anObject
    bm1 := anObject!

```

```

bm2
    ^bm2!

```

```

bm2: anObject
    bm2 := anObject!

```

bm: anObject
bm := anObject!

buffs1
^buffs1!

buffs1: anObject
buffs1 := anObject!

buffs2
^buffs2!

buffs2: anObject
buffs2 := anObject!

buffs3
^buffs3!

buffs3: anObject
buffs3 := anObject!

effectif
^effectif!

effectif: anObject
effectif := anObject!

film
^film!

film: anObject
film := anObject!

s1
^s1!

s1: anObject
s1 := anObject!

s2
^s2!

s2: anObject
s2 := anObject!

s3
^s3!

s3: anObject
s3 := anObject!

xi1
^xi1!

xi1: anObject
xi1 := anObject!

xi2
^xi2!

xi2: anObject
xi2 := anObject! !

!Biofilm_Cell methodsFor: 'pov'!

pdvAc

self s2 isNil ifTrue: [#tresfaible].
self s2 <0.1 ifTrue: [#tresfaible].
self s2 <0.25 ifTrue: [#faible].
self s2 <0.5 ifTrue: [#moyen].
self s2 <0.75 ifTrue: [#moyen1].
self s2 <1 ifTrue: [#moyen2].
self s2 <1.25 ifTrue: [#moyen3].
self s2 <1.5 ifTrue: [#moyen4].
self s2 <2 ifTrue: [#fort].
^#fort.!

pdVBiofilm

```
self film ifTrue:[^#film].
self film ifFalse:[^#liquide].!
```

pdvSubstrat

```
self s1 isNil ifTrue: [^#tresfaible].
self s1 <5 ifTrue: [^#tresfaible].
self s1 <10 ifTrue: [^#faible].
self s1 <15 ifTrue: [^#moyen].
self s1 <16 ifTrue: [^#moyen1].
self s1 <17 ifTrue: [^#moyen2].
self s1 <18 ifTrue: [^#moyen3].
self s1 <19 ifTrue: [^#moyen4].
self s1 <20 ifTrue: [^#fort].
^#tresfort.!
```

povbm1

```
self bm1 <1 ifTrue: [^#faible].
self bm1 <5 ifTrue: [^#moyen].
self bm1 <10 ifTrue: [^#moyen1].
self bm1 <15 ifTrue: [^#moyen2].
self bm1 <20 ifTrue: [^#moyen3].
self bm1 <25 ifTrue: [^#moyen4].
self bm1 <30 ifTrue: [^#fort].
self bm1 <40 ifTrue: [^#tresfort].
^#tresfort.!
```

povbm2

```
self bm2 <1 ifTrue: [^#faible].
self bm2 <5 ifTrue: [^#moyen].
self bm2 <10 ifTrue: [^#moyen1].
self bm2 <15 ifTrue: [^#moyen2].
self bm2 <20 ifTrue: [^#moyen3].
self bm2 <25 ifTrue: [^#moyen4].
self bm2 <30 ifTrue: [^#fort].
self bm2<40 ifTrue: [^#tresfort].
^#tresfort.!
```

povSeuil

```
((self bm1+self bm2) <(BMax/2)) ifTrue:[^#vide].
((self bm1+self bm2) <(0.9*BMax)) ifTrue:[^#plein].
^#saturation.!
```

povSubstrat

```
self s1 isNil ifTrue: [^#tresfaible].
self s1 <5 ifTrue: [^#tresfaible].
self s1 <10 ifTrue: [^#faible].
self s1 <15 ifTrue: [^#moyen].
self s1 <20 ifTrue: [^#fort].
^#fort.!!
```

Cormas_Model subclass: #Biofilm

```

instanceVariableNames: 'theBiofilm_Cells theBiofilm_Colonies effectif substrat acetate X1 X2 theBiofilm_Acetos
theBiofilm_Methanos mlap '
classVariableNames: "
poolDictionaries: "
category: 'Cormas-Biofilm!'

```

!Biofilm methodsFor: 'instance-creation'!**initAgents**

```

"Cr ation et Installation des agents sur la grille"
| colonie bord rang |
super initAgents.

```

```

"Selection de la zone a ensemencher"

```

```

bord:=OrderedCollection new.
rang:=0.
2 timesRepeat: [
    theBiofilm_Cells do: [:cellule] (((cellule id+rang) rem: self space colonne) =0) ifTrue:[bord add:cellule]].
rang:=(rang+1)].

```

```

"Distribution des agents"

```

```

bord do:[:cellule]    colonie := Biofilm_Aceto new.
                    colonie init.
                    colonie moveTo: cellule.
                    colonie patch film:true.
                    "liste les cellules entre la colonie et la phase liquide"
                    self rayon: colonie.
                    theBiofilm_Acetos add: colonie.
                    theBiofilm_Colonies add:colonie.].

```

```

bord do:[:cellule]    colonie := Biofilm_Methano new.
                    colonie init.
                    colonie moveTo: cellule.
                    colonie patch film:true.
                    "liste les cellules entre la colonie et la phase liquide"
                    self rayon: colonie.
                    theBiofilm_Methanos add: colonie.
                    theBiofilm_Colonies add:colonie.].!

```

initAgents2

```

"Cr ation et Installation des agents sur la grille"
| colonie bord rang |
super initAgents.

```

```

"Selection de la zone a ensemencher"

```

```

bord:=OrderedCollection new.
rang:=0.
1 timesRepeat: [
    theBiofilm_Cells do: [:cellule] (((cellule id+rang) rem: self space colonne) =0) ifTrue:[bord add:cellule]].
rang:=(rang+1)].

```

```

"Distribution des agents"

```

```

bord do:[:cellule]    colonie := Biofilm_Aceto new.
                    colonie init.
                    colonie moveTo: cellule.
                    colonie patch film:true.
                    "liste les cellules entre la colonie et la phase liquide"
                    self rayon: colonie.
                    theBiofilm_Acetos add: colonie.
                    theBiofilm_Colonies add:colonie.].

```

```

bord do:[:cellule]    colonie := Biofilm_Methano new.
                    colonie init.
                    colonie moveTo: cellule.
                    colonie patch film:true.
                    "liste les cellules entre la colonie et la phase liquide"
                    self rayon: colonie.
                    theBiofilm_Methanos add: colonie.
                    theBiofilm_Colonies add:colonie.].!!

```

```

!Biofilm methodsFor: 'init'!

```

```

init

```

```

"ParamÃtres rÃacteur"
Ac0:=(0.7).
S0:=(2.2).
Met0:=0. "debit massique de methane en gl-1j-1"
Epaisseur:=0. "epaisseur moyenne du biofilm en m"
X1r:=0. "concentration moyenne ds le reacteur en acetos en g/L"
X2r:=0.

Acln:=(1).
Sin:=(10). "g/L"

TSH:=(1). "jour"
SurfSpe:=(180). "surface specifique mÃ²/m3"

"ParamÃtres simulation"
Dt:=(1e-3). "j"
DI:=50e-6. "m soit 50Ãµm"
Df:=50e-6.
X10:=(5*(DI*DI*Df)*0.001). "g/L"
X20:=(5*(DI*DI*Df)*0.001)."2.5"
"ParamÃtre biofilm"

Thickness:=400e-6. "epaisseur en nb cellules=Ãµm/DI"
Gamma:=(0.3)."0.39."
D1DCO:=(0.6e-4). "mÃ²/j"
D2DCO:=(0.6e-4).
D1Ac:=(1.03e-4).
D2Ac:=(1.03e-4).
KFeed:=5.
BMax:=30. "g/L"
Bseuil:=(1*BMax/30*(DI*DI*Df)*0.001). "g"
EffSup:=5.

"ParamÃtres biologiques"
Ks1:=(7.1). "g/L"
Ks2:=(0.55).
Ki2:=(0.96). "(g/L)"
Mu1Max:=(1.2). "jour^-1"
Mu2Max:=(0.74).

k11:=-42.
k12:=(6.99).
k21:=(-16.09).
k22:=(5.04).
Lig:=self space ligne.
Col:=self space colonne.
self initCells.
self initAgents.

self writeParam.
theBiofilm_Colonies do: [:colonie]colonie calculMu].
self bilanModele.

mlap:=VensimDde new.!

init1bis
"ParamÃtres rÃacteur"
Ac0:=(0.7).
S0:=(2.2).
Met0:=0. "debit massique de methane en gl-1j-1"
Epaisseur:=0. "epaisseur moyenne du biofilm en m"
X1r:=0. "concentration moyenne ds le reacteur en acetos en g/L"
X2r:=0.

Acln:=(2).
Sin:=(10). "g/L"

TSH:=(1). "jour"
SurfSpe:=(180). "surface specifique mÃ²/m3"

"ParamÃtres simulation"
Dt:=(1e-3). "j"
DI:=10e-6. "m soit 50Ãµm"
Df:=50e-6.
X10:=(5*(DI*DI*Df)*0.001). "g/L"

```

```
X20:=(5*(DI*DI*Df)*0.001)."2.5"
"ParamÃtre biofilm"
```

```
Thickness:=800e-6.          "epaisseur en nb cellules=Åµm/DI"
Gamma:=(0.3)."0.39."
D1DCO:=(0.6e-4).          "mÅ²/j"
D2DCO:=(0.6e-4).
D1Ac:=(1.03e-4).
D2Ac:=(1.03e-4).
KFeed:=5.
BMax:=30.                  "g/L"
Bseuil:=(1*BMax/30*(DI*DI*Df)*0.001). "g"
EffSup:=10.
```

```
"ParamÃtres biologiques"
Ks1:=(7.1).                "g/L"
Ks2:=(0.55).
Ki2:=(0.96).              "(g/L)"
Mu1Max:=(1.2).            "jour^-1"
Mu2Max:=(0.74).
```

```
k11:=-42.
k12:=(6.99).
k21:=(-16.09).
k22:=(5.04).
Lig:=self space ligne.
Col:=self space colonne.
      self initCells.
      self initAgents.
```

```
self writeParam.
theBiofilm_Colonies do: [:colonie|colonie calculMu].
self bilanModele.!!
```

!Biofilm methodsFor: 'procedure'!

bilanModele

```
"Calcule les variables d'etat du reacteur"
```

```
| methane bf |
```

```
bf:=0.
X1r:=0.
X2r:=0.
methane:=0.
```

```
theBiofilm_Cells do:      [:cellule] cellule bilanBM.].
theBiofilm_Cells do:      [:cellule] cellule bilanFilm.
```

```
[bf:=bf+1].
```

```
(cellule bm>0) ifTrue:
```

```
X1r:=X1r+cellule bm1.
X2r:=X2r+cellule bm2.
methane:=methane+cellule
```

```
s3].
```

```
"epaisseur du biofilm"
Epaisseur:=DI*bf/Lig.
```

```
"debit mÃ©thane L/Lreac/h"
Met0:=methane*SurfSpe*DI/(Lig*Dt*24)/16*25.
```

```
"concentrations en biomasse moyenne vues du reacteur"
X1r:=X1r*SurfSpe*DI/Lig.
X2r:=X2r*SurfSpe*DI/Lig.!
```

connectReactor:repet a:t

```
"Effet de l'alimentation du reacteur et de l'activitÃ© du biofilm sur la concentration dans la phase liquide"
|boundLayer sub ac s a |
```

```
"Determine le flux entre biofilm et phase liquide hyp que flux = f(dS de la couche limite)"
"Selectionne les cellules de la couche limite"
boundLayer:= OrderedCollection new.
boundLayer:=theBiofilm_Cells select:[:cellule| (cellule film) and:[:(cellule bm1 + cellule bm2)=0]].
"Fait la moyenne des dS le long de l'interface"
```

```

sub:=0. ac:=0.
boundLayer do:[cellule| sub:=sub+(cellule s1). ac:=ac+(cellule s2).].
sub:=sub/(boundLayer size).
ac:=ac/(boundLayer size).

```

```

sub:=(S0-sub)/Dl*D2DCO.
ac:=(Ac0-ac)/Dl*D2Ac.
"Determine les nvlles concentrations de la phase liquide"
s:= (((Sin-S0)/TSH) - (SurfSpe*sub))*Dt/repet.
a:=(((AcIn-Ac0)/TSH)-(SurfSpe*ac))*Dt/repet.
S0:=S0+s". Ac0:=Ac0+a".

```

```

theBiofilm_Cells do: [cellule| (cellule film) iffFalse:[cellule s1:S0. cellule s2:Ac0]].!

```

controleur1:but

```

"simule l'effet d'un controleur sur le modÃ"le"
"controleur d'apaptation des biomasses INRIA"

```

```

| Qin inf sup Vm met |

```

```

Vm:=948. "volume reacteur reel en L"
met:=Met0*Vm/24. "dÃ©bit CH4 L/h"
inf:=25.
sup:=50.

```

```

Qin:=3.8*met/(Sin+(1.48*AcIn)-but).
(Qin<inf) iffTrue:[Qin:=inf].
(Qin>sup) iffTrue:[Qin:=sup].

```

```

TSH:=(Vm/Qin/24) asFloat. "temps de sÃ©jour en jour"!

```

controleur2

```

"controle le debit d'alim pour maintenir la concentration de substrat"

```

```

(S0>(Sin-1)) iffTrue: [TSH:=TSH*1.1].
(S0<1.5) iffTrue: [TSH:=TSH/1.1].

```

```

TSH:=TSH max: 0.8.
TSH:=TSH min: 7.!

```

nettoyage:num

```

"arrondit toutes les variables Ã la puissance de num"

```

```

theBiofilm_Cells do: [cellule|

```

```

self tronque:cellule s1 at:num.
self tronque:cellule s2 at:num.
(cellule s1<0) iffTrue: [cellule s1:0].
(cellule s2<0) iffTrue: [cellule s2:0]].

```

```

theBiofilm_Colonies do:[colonie|

```

```

self tronque:colonie m at:num.
(colonie m<0) iffTrue:[colonie m:0].!

```

rayon:colonie

```

"determine les cellules comprises entre la colonie et la phase liquide"

```

```

| limite |

```

```

theBiofilm_Cells do: [cellule| limite:=colonie patch id- ((colonie patch id -1) rem: self space colonie).

```

```

((cellule id>=limite ) and: [cellule

```

```

id<colonie patch id]) iffTrue: [colonie rayon add: cellule]].!

```

sauveDemographie: nomFichier num: ordre

```

| newFile stream |

```

```

newFile := (nomFichier,'A', ordre printString, '.txt') asFilename.

```

```

stream := newFile writeStream.

```

```

self theBiofilm_Acetos do:

```

```

[a | stream nextPutAll: a m printString.

```

```

stream nextPutAll: ' '.

```

```

stream nextPutAll: a age printString.

```

```

stream nextPutAll: ' '.

```

```

stream nextPutAll: a patch id printString, '\ withCRs].

```

```

stream close.

```

```

newFile := (nomFichier,'B', ordre printString, '.txt') asFilename.

```

```

stream := newFile writeStream.
self theBiofilm_Methanos do:
    [:a | stream nextPutAll: a m printString.
        stream nextPutAll: ' '.
        stream nextPutAll: a age printString.
        stream nextPutAll: ' '.
        stream nextPutAll: a patch id printString, '\ withCRs].
stream close.!

```

sauveEnvironnement: nomFichier

```

| newFile stream |
newFile := nomFichier asFilename.
stream := newFile writeStream.
self theBiofilm_Cells do: [:a |
    a s1: a s1 asFloat."
"
    a s2: a s2 asFloat."
    stream nextPutAll: a s1 printString.
    stream nextPutAll: ' '.
    stream nextPutAll: a s2 printString.
    stream nextPutAll: ' '.
    stream nextPutAll: a s3 printString.
    stream nextPutAll: ' '.
    stream nextPutAll: (a bm1) printString.
    stream nextPutAll: ' '.
    stream nextPutAll: (a bm2) printString.
    stream nextPutAll: ' '.
    stream nextPutAll: (a effectif) printString, '\ withCRs].

stream close.
"theBiofilmColonies do: [:colonie]"
"
    colonie x1: colonie x1 asFloat."
"
    colonie x2: colonie x2 asFloat.]"!

```

sauveEtat: nomFichier

```

"sauve les variables d'etat du reacteur"
| newFile stream |
newFile := nomFichier asFilename.
stream := newFile writeStream.
stream nextPutAll: S0 printString, '\ withCRs.
stream nextPutAll: Ac0 printString, '\ withCRs.
stream nextPutAll: Met0 printString, '\ withCRs.
stream nextPutAll: TSH printString, '\ withCRs .
stream nextPutAll: Epaisseur printString, '\ withCRs .
stream nextPutAll: X1r printString, '\ withCRs.
stream nextPutAll: X2r printString, '\ withCRs.
stream close.!

```

stepMatlab2:theCellules lig:l col:n

```

"Appelle matlab pour le calcul charge-diffusion-reaction"
"Stocke dans les buffers les influences de chacun des processus qui feront la valeur du temps t+1"

```

```

| "mlap" result grille mot matrice requete |

```

```

"Preparation des donnees"
matrice:=['.
theCellules do:[:cellule]
matrice:=matrice,(((cellule s1) reduceGeneralityIfPossible)
literalArrayEncoding) printString, ' '.
matrice:=matrice,(((cellule s2)reduceGeneralityIfPossible)
literalArrayEncoding ) printString, ';' ].
matrice:=matrice, ']' .
mlap execute: 'a=',matrice.

matrice:=['.
theCellules do:[:cellule]
matrice:=matrice,(((cellule xi1)reduceGeneralityIfPossible)
literalArrayEncoding ) printString, ' '.
matrice:=matrice,(((cellule xi2)reduceGeneralityIfPossible)
literalArrayEncoding ) printString, ';' ].
matrice:=matrice, ']' .
mlap execute: 'b=',matrice.

requete:='reponse=diffusionR([a,b],',Lig printString,',',Col printString,',',Dt printString,',',((S0*100) truncated)/100)
printString,',',(((Ac0*100) truncated)/100) printString,')'.

```



```

        mlap execute: requete.
        result:=mlap request: 'reponse'.
"
        mlap terminate."
"
        self halt."
"result est la matrice reponse sous forme de ByteString"

        grille:=OrderedCollection new.
        mot:=".
"place les valeurs numériques dans la collection grille"
        result do:[:car] car isSeparator
                                ifFalse:[mot:=mot copyWith:(car)]
                                ifTrue: [ (mot size=0) ifFalse:[grille add:(mot asNumber)].
                                        mot:="].].

theCellules do:[:cellule]      cellule s1: (grille at:((cellule id)*3-2)) asDouble.
                                cellule s2: (grille at:((cellule id)*3-1)) asDouble.
                                cellule s3: (grille at:((cellule id)*3)) asDouble.].!

stepMatlab2:theCellules lig:l col:n fois:f
"Appelle matlab pour le calcul charge-diffusion-reaction"
"Stocke dans les buffers les influences de chacun des processus qui feront la valeur du temps t+1"

| "mlap" result grille mot matrice requete |

"Preparation des donnees"
        matrice:=['.
        theCellules do:[:cellule]
                                matrice:=matrice,(((cellule s1) reduceGeneralityIfPossible)
literalArrayEncoding) printString,' '.
                                matrice:=matrice,(((cellule s2)reduceGeneralityIfPossible)
literalArrayEncoding ) printString,' '.
                                matrice:=matrice,(((cellule xi1)reduceGeneralityIfPossible)
literalArrayEncoding ) printString,' '.
                                matrice:=matrice,(((cellule xi2)reduceGeneralityIfPossible)
literalArrayEncoding ) printString,'; ].
        matrice:=matrice, ']'.

"
        mlap:=VensimDde new."
        mlap execute: 'a=',matrice.
requete:=reponse=diffusionR2(a,',Lig printString,',',Col printString,',',Dt printString,',',((S0*100) truncated)/100)
printString,',',(((Ac0*100) truncated)/100) printString,',', f printString,')'.
        mlap execute: requete.
        result:=mlap request: 'reponse'.
        mlap terminate.
"
        self halt."
"result est la matrice reponse sous forme de ByteString"

        grille:=OrderedCollection new.
        mot:=".
"place les valeurs numériques dans la collection grille"
        result do:[:car] car isSeparator
                                ifFalse:[mot:=mot copyWith:(car)]
                                ifTrue: [ (mot size=0) ifFalse:[grille add:(mot asNumber)].
                                        mot:="].].

theCellules do:[:cellule]      cellule s1: (grille at:((cellule id)*3-2)) asDouble.
                                cellule s2: (grille at:((cellule id)*3-1)) asDouble.
                                cellule s3: (grille at:((cellule id)*3)) asDouble.].!

```

stepMatlab:theCellules lig:l col:n

"Appelle matlab pour le calcul charge-diffusion-reaction"

"Stocke dans les buffers les influences de chacun des processus qui feront la valeur du temps t+1"

| "mlap" result grille mot matrice requete |

"Preparation des donnees"

```

        matrice:=['.
        theCellules do:[:cellule]

```

```

                                matrice:=matrice,(((cellule s1) reduceGeneralityIfPossible)

```

```

literalArrayEncoding) printString,' '.

```

```

                                matrice:=matrice,(((cellule s2)reduceGeneralityIfPossible)

```

```

literalArrayEncoding ) printString,' '.

```

```

literalArrayEncoding ) printString, ' '.
matrix:=matrice,(((cellule xi1)reduceGeneralityIfPossible)

literalArrayEncoding ) printString, '; '].
matrix:=matrice, ''].

"
  mlap:=VensimDde new."
  mlap execute: 'a=',matrice.
  requete:=reponse=diffusionR(a,',Lig printString,',',Col printString,',',Dt printString,',',(((S0*100) truncated)/100)
printString,',',(((Ac0*100) truncated)/100) printString,').
  mlap execute: requete.
  result:=mlap request: 'reponse'.
  mlap terminate."
"
  self halt."
"result est la matrice reponse sous forme de ByteString"

  grille:=OrderedCollection new.
  mot:=".
"place les valeurs numÃ©riques dans la collection grille"
  result do:[:car| car isSeparator
    ifFalse:[mot:=mot copyWith:(car)]
    ifTrue: [ (mot size=0) ifFalse:[grille add:(mot asNumber)].
              mot=""].

theCellules do:[:cellule|
  cellule s1: (grille at:((cellule id)*3-2)) asDouble.
  cellule s2: (grille at:((cellule id)*3-1)) asDouble.
  cellule s3: (grille at:((cellule id)*3)) asDouble.].

tronque:num at:puissance
"arrondi un nombre"

^((num*puissance) truncated)/puissance.!

writeParam
"Ecriture d'un fichier de paramÃ©tres"
| newFile stream |
  newFile := 'c:\vwnc30\cormas\data\param.txt' asFilename.
  stream := newFile writeStream.
  stream nextPutAll: S0 printString, '\ withCRs.
  stream nextPutAll: Sin printString, '\ withCRs.
  stream nextPutAll: Ac0 printString, '\ withCRs.
  stream nextPutAll: Ac1n printString, '\ withCRs.
  stream nextPutAll: TSH printString, '\ withCRs.
  stream nextPutAll: SurfSpe printString, '\ withCRs.
  stream nextPutAll: D1DCO printString, '\ withCRs.
  stream nextPutAll: D1Ac printString, '\ withCRs.
  stream nextPutAll: DI printString, '\ withCRs.
  stream nextPutAll: Ks1 printString, '\ withCRs.
  stream nextPutAll: k11 printString, '\ withCRs.
  stream nextPutAll: k12 printString, '\ withCRs.
  stream nextPutAll: Ks2 printString, '\ withCRs.
  stream nextPutAll: k21 printString, '\ withCRs.
  stream nextPutAll: k22 printString, '\ withCRs.
  stream nextPutAll: Ki2 printString, '\ withCRs.
  stream nextPutAll: Mu1Max printString, '\ withCRs.
  stream nextPutAll: Mu2Max printString, '\ withCRs.
  stream nextPutAll: Lig printString, '\ withCRs.
  stream nextPutAll: Col printString, '\ withCRs.
  stream nextPutAll: X10 printString, '\ withCRs.
  stream nextPutAll: X20 printString, '\ withCRs.
  stream nextPutAll: BMax printString, '\ withCRs.
  stream nextPutAll: Bseuil printString, '\ withCRs.
  stream nextPutAll: KFeed printString, '\ withCRs.
  stream nextPutAll: EffSup printString, '\ withCRs.
  stream nextPutAll: Thickness printString, '\ withCRs.
  stream nextPutAll: Dt printString, '\ withCRs.

  stream close!

writeParam:N
"Ecriture d'un fichier de paramÃ©tres"
| newFile stream |
  newFile := ('c:\vwnc30\cormas\param', (N printString), '.txt') asFilename.
  stream := newFile writeStream.

```

```

stream nextPutAll: S0 printString, '\ withCRs.
stream nextPutAll: Sin printString, '\ withCRs.
stream nextPutAll: Ac0 printString, '\ withCRs.
stream nextPutAll: AcIn printString, '\ withCRs.
stream nextPutAll: TSH printString, '\ withCRs.
stream nextPutAll: SurfSpe printString, '\ withCRs.
stream nextPutAll: D1DCO printString, '\ withCRs.
stream nextPutAll: D1Ac printString, '\ withCRs.
stream nextPutAll: DI printString, '\ withCRs.
stream nextPutAll: Ks1 printString, '\ withCRs.
stream nextPutAll: k11 printString, '\ withCRs.
stream nextPutAll: k12 printString, '\ withCRs.
stream nextPutAll: Ks2 printString, '\ withCRs.
stream nextPutAll: k21 printString, '\ withCRs.
stream nextPutAll: k22 printString, '\ withCRs.
stream nextPutAll: Ki2 printString, '\ withCRs.
stream nextPutAll: Mu1Max printString, '\ withCRs.
stream nextPutAll: Mu2Max printString, '\ withCRs.
stream nextPutAll: Lig printString, '\ withCRs.
stream nextPutAll: Col printString, '\ withCRs.
stream nextPutAll: X10 printString, '\ withCRs.
stream nextPutAll: X20 printString, '\ withCRs.
stream nextPutAll: BMax printString, '\ withCRs.
stream nextPutAll: Bseuil printString, '\ withCRs.
stream nextPutAll: KFeed printString, '\ withCRs.

```

```
stream close! !
```

!Biofilm methodsFor: 'control'!

step1: t

```
"Pas de temps du modele triphasÃ© : liq+BL+BF"
```

```
| theAgents repet|
```

```
"Boucle de diffusion-reaction"
```

```
repet:=250.
```

```
repet timesRepeat: [
```

```
    "diffusion pour les cellules du biofilm et de la couche limite"
```

```
    (theBiofilm_Cells select:[:cellule|(cellule film)]) do:[:cellule|cellule step:repet.].
```

```
    "echange avec le reacteur"
```

```
    self connectReactor:repet a:t.
```

```
    "MaJ changements "
```

```
    theBiofilm_Cells do: [:cellule| cellule influence]].
```

```
"Dynamique bacterienne"
```

```
theAgents:=OrderedCollection new.
```

```
theBiofilm_Colonies do: [:colonie| colonie step: theAgents.].
```

```
(theAgents size >0) ifTrue: [theAgents do: [:colonie | theBiofilm_Colonies add:colonie.
```

```
self rayon:colonie]].
```

```
theBiofilm_Colonies do: [:colonie| colonie influence].
```

```
theBiofilm_Colonies do: [:colonie| (colonie dead) ifTrue: [colonie leave.]].
```

```
theBiofilm_Acetos removeAllSuchThat: [:colonie|(colonie dead) ].
```

```
theBiofilm_Methanos removeAllSuchThat: [:colonie|(colonie dead) ].
```

```
theBiofilm_Colonies removeAllSuchThat: [:colonie|(colonie dead) ].
```

```
"Mise a jour"
```

```
theBiofilm_Cells do: [:cellule| cellule bilanBM.].
```

```
theBiofilm_Cells do: [:cellule| cellule bilanFilm.].
```

```
"Sauvegarde reguliere de la grille spatiale"
```

```
(t rem: 50)=0 ifTrue: [self sauveEnvironnement: 'env', t printString, '.txt'.
```

```
self sauveDemographie: 'population',t printString, '.txt'.
```

```
self nettoyage:100000].
```

```
" self updateCharts: t!"
```

step1bis: t

```
"Pas de temps du modele triphasÃ© : liq+BL+BF"
```

```
| theAgents repet|
```

```
"Boucle de diffusion-reaction"
```

```
repet:=250*10.
```

```

repet timesRepeat: [
    "diffusion pour les cellules du biofilm et de la couche limite"
    (theBiofilm_Cells select:[cellule|(cellule film)]) do:[cellule|cellule step:repet.].
    "echange avec le reacteur"
    self connectReactor:repet a:t.
    "MaJ changements "
    theBiofilm_Cells do: [:cellule| cellule influence]].

"Dynamique bacterienne"
theAgents:=OrderedCollection new.
theBiofilm_Colonies do: [:colonie| colonie step: theAgents.].
(theAgents size >0) ifTrue: [theAgents do: [:colonie | theBiofilm_Colonies add:colonie.

    self rayon:colonie]].

theBiofilm_Colonies do: [:colonie| colonie influence].

theBiofilm_Colonies do: [:colonie| (colonie dead) ifTrue: [colonie leave.]].
theBiofilm_Acetos removeAllSuchThat: [:colonie|(colonie dead) ].
theBiofilm_Methanos removeAllSuchThat: [:colonie|(colonie dead) ].
theBiofilm_Colonies removeAllSuchThat: [:colonie|(colonie dead) ].

"Mise a jour"
self bilanModele.
self controleur2.

"Sauvegarde reguliere de la grille spatiale"
(t rem: 50)=0 ifTrue: [self sauveEnvironnement: 'C:\vwnc30\cormas\data\env', t printString, '.txt'.
    self sauveDemographie: 'C:\vwnc30\cormas\data\population' num:
t.

    self sauveEtat: 'C:\vwnc30\cormas\data\etat',t printString, '.txt'.
    self nettoyage:100000].!

step2: t
"Pas de temps du modele triphasÃ© : liq+BL+BF"

| theAgents repet|

"Boucle de diffusion-reaction"
repet:=1.
repet timesRepeat: [
    "diffusion pour les cellules du biofilm et de la couche limite"
    self stepMatlab2: theBiofilm_Cells lig: Lig col:Col.
    "echange avec le reacteur"
    self connectReactor:repet a:t.].

"Dynamique bacterienne"
theAgents:=OrderedCollection new.
theBiofilm_Colonies do: [:colonie| colonie step: theAgents.].
(theAgents size >0) ifTrue: [theAgents do: [:colonie | theBiofilm_Colonies add:colonie.

    self rayon:colonie.
].self halt.].

theBiofilm_Colonies do: [:colonie| colonie influence].

theBiofilm_Colonies do: [:colonie| (colonie dead) ifTrue: [colonie leave.]].
theBiofilm_Acetos removeAllSuchThat: [:colonie|(colonie dead) ].
theBiofilm_Methanos removeAllSuchThat: [:colonie|(colonie dead) ].
theBiofilm_Colonies removeAllSuchThat: [:colonie|(colonie dead) ].

"Mise a jour"
self bilanModele.
self controleur2."

"Sauvegarde reguliere de la grille spatiale"
(t rem: 50)=0 ifTrue: [self sauveEnvironnement: 'C:\vwnc30\cormas\data\env', t printString, '.txt'.
    self sauveDemographie: 'C:\vwnc30\cormas\data\population' num:
t.

```

```
self sauveEtat: 'C:\vwnc30\cormas\data\etat', t printString, '.txt'.
self nettoyage:100000].!
```

step2: t fois:N

```
"Pas de temps du modele triphasÃ© : liq+BL+BF"
```

```
| theAgents repet|
```

```
(t=0) ifTrue: [
(N=1) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=1. X10:=5. X20:=5].
(N=2) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=5. X10:=5. X20:=5].
(N=3) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=10. X10:=5. X20:=5].
(N=4) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=1. X10:=1. X20:=1].
(N=5) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=1. X10:=10. X20:=10].
(N=6) ifTrue: [Sin:=10. Acln:=2. BMax:=30. Bseuil:=1. X10:=30. X20:=30].
(N=7) ifTrue: [Sin:=2. Acln:=0. BMax:=30. Bseuil:=1. X10:=5. X20:=5].
(N=8) ifTrue: [Sin:=2. Acln:=0.5. BMax:=30. Bseuil:=1. X10:=5. X20:=5].
(N=9) ifTrue: [Sin:=5. Acln:=0. BMax:=30. Bseuil:=1. X10:=5. X20:=5].
(N=10) ifTrue: [Sin:=16. Acln:=3. BMax:=30. Bseuil:=1. X10:=5. X20:=5].
```

```
self writeParam:N].
```

```
"Boucle de diffusion-reaction"
```

```
repet:=1.
repet timesRepeat: [
"diffusion pour les cellules du biofilm et de la couche limite"
self stepMatlab2: theBiofilm_Cells lig: Lig col:Col fois:N.
"echange avec le reacteur"
self connectReactor:repet a:t.].
```

```
"Dynamique bacterienne"
```

```
theAgents:=OrderedCollection new.
theBiofilm_Colonies do: [:colonie| colonie step: theAgents.].
(theAgents size >0) ifTrue: [theAgents do: [:colonie | theBiofilm_Colonies add:colonie.
```

```
self rayon:colonie]].
```

```
theBiofilm_Colonies do: [:colonie| colonie influence].
```

```
theBiofilm_Colonies do: [:colonie| (colonie dead) ifTrue: [colonie leave.]].
theBiofilm_Acetos removeAllSuchThat: [:colonie| (colonie dead) ].
theBiofilm_Methanos removeAllSuchThat: [:colonie| (colonie dead) ].
theBiofilm_Colonies removeAllSuchThat: [:colonie| (colonie dead) ].
```

```
"Mise a jour"
```

```
self bilanModele.
self controleur2.
```

```
"Sauvegarde reguliere de la grille spatiale"
```

```
(t rem: 50)=0 ifTrue: [self sauveEnvironnement: 'C:\vwnc30\cormas\data', (N printString), 'env', t printString, '.txt'.
self sauveDemographie: 'C:\vwnc30\cormas\data', (N
printString), 'population' num: t.
self sauveEtat: 'C:\vwnc30\cormas\data', (N printString), 'etat', t
printString, '.txt'.
self nettoyage:100000].!
```

!Biofilm methodsFor: 'accessing'!

```
theBiofilm_Acetos
^theBiofilm_Acetos!
```

```
theBiofilm_Acetos: x
theBiofilm_Acetos := x!
```

```
theBiofilm_Cells
^theBiofilm_Cells!
```

```
theBiofilm_Cells: x
theBiofilm_Cells := x!
```

```
theBiofilm_Colonies
^theBiofilm_Colonies!
```

```
theBiofilm_Colonies: anObject
```

```

theBiofilm_Colonies := anObject!

theBiofilm_Methanos
^theBiofilm_Methanos!

theBiofilm_Methanos: x
theBiofilm_Methanos := x! !

!Biofilm methodsFor: 'graphiques'!

acetate
^acetate isNil
ifTrue: [acetate := List new]
ifFalse: [acetate]!

acetate: x
acetate := x!

acetateData
"return the data (a number) to be plotted with the acetate chart"
|total|
total:=0.
theBiofilm_Cells do:[:cellule|total:=cellule s2 + total].
^total!

effectif
^effectif isNil
ifTrue: [effectif := List new]
ifFalse: [effectif]!

effectif: x
effectif := x!

effectifData
"return the data (a number) to be plotted with the effectif chart"
^theBiofilmColonies size!

substrat
^substrat isNil
ifTrue: [substrat := List new]
ifFalse: [substrat]!

substrat: x
substrat := x!

substratData
"return the data (a number) to be plotted with the substrat chart"
|total|
total:=0.
theBiofilm_Cells do:[:cellule|total:=cellule s1 + total].
^total!

X1
^X1 isNil
ifTrue: [X1 := List new]
ifFalse: [X1]!

X1: x
X1 := x!

X1Data
"return the data (a number) to be plotted with the X1 chart"
| total |
total:=0.
theBiofilm_Cells do:[:cellule|total:=total+cellule bm1].
^total!

X2
^X2 isNil
ifTrue: [X2 := List new]
ifFalse: [X2]!

X2: x
X2 := x!

X2Data

```

```
"return the data (a number) to be plotted with the X2 chart"  
| total |  
    total:=0.  
theBiofilm_Cells do[:cellule|total:=total+cellule bm2].  
^total!
```

```

AgentLocation subclass: #Biofilm_Colonie
  instanceVariableNames: 'm pere dynastie mu decay death buffm rayon age muM '
  classVariableNames: 'CurrentId '
  poolDictionaries: ''
  category: 'Cormas-Biofilm!'

```

!Biofilm_Colonie methodsFor: 'init'!

init

```

super init.
self m: 0.
self buffm: 0.
self mu: 0.
self mu: 0.
self death: 0.
self age: 0.
self dead: false.
self dynastie: OrderedCollection new.
self rayon: OrderedCollection new.!!

```

!Biofilm_Colonie methodsFor: 'procedure'!

arrachement

```

"Arrachement du biofilm des colonies en inanition depuis 48h"
"on enlève la bactérie en inanition, ses filles du voisinages et ttes les colonies entre elle et la phase liquide"

```

| wagon |

```

self dead: true.
wagon:=OrderedCollection new.
self rayon do:[:cellule| cellule theOccupants do:[:membre| membre do:[:colonie|
"(self dynastie includes:colonie) ifTrue:["wagon add:colonie"]]]].
wagon do: [:colonie| colonie dead: true].
"self halt. "!

```

arrachement2

```

"Arrachement du biofilm des colonies en inanition depuis 48h"
"on enlève la bactérie en inanition, ses filles du voisinages et ttes les colonies entre elle et la phase liquide"

```

```

self dead: true. !

```

croissance:lesColonies

| dm voisin liste |

```

"Augmentation des effectifs de la colonie : les buffX contiennent le dX"
dm:=(self mu-self decay)* self m*Dt.

```

```

((dm<0) or: (((self patch bm < BMax) and: [ m<Bseuil]))) ifTrue: [self buffm:self buffm+dm].
((dm>0) and: (((self patch bm > (Bseuil/(DI*DI*Df*0.001))) or: [ m>Bseuil]))) ifTrue: [

```

```

"liste les cellules voisines avec de la place et dans l'epaisseur autorisée de biofilm"

```

```

voisin:=OrderedCollection new.
self patch neighbourhood do:[:cellule| (cellule bm<BMax ) ifTrue: [voisin add:cellule]].

```

```

voisin removeAllSuchThat: [:cellule| ((Col-(cellule id rem:Col)>(Thickness/DI) and:[(cellule id rem:Col)>0])). ].

```

```

(voisin size=0) ifTrue: ["pas de voisin libre"

```

```

(self patch bm<BMax) ifTrue: [self buffm:self buffm+dm]

```

```

ifFalse: [self buffm:self buffm]

```

```

ifFalse: ["voisins libres"

```

```

liste:=OrderedCollection new.

```

```

self dynastie do:[:fille|(voisin includes: fille patch) ifTrue: [liste add:

```

```

fille. voisin remove: fille patch]].

```

```

"division cellulaire"

```

```

voisin do:[:cellule| self division:cellule dans:lesColonies vers:liste].

```

```

"transfert de biomasse vers les filles"

```

```

(liste size>0) ifTrue: [liste do:[:colonie| colonie buffm: colonie

```

```

buffm+(dm/liste size) ].].

```

```

].

```

```

].!!

```

!Biofilm_Colonie methodsFor: 'control'!

influence


```
"met a jour les variables avec leur buffer"
self m: (self m + self buffm) asDouble.
"met les buffer a zero"
self buffm:0 asDouble.
```

```
"elimine les colonies a concentration nÃ©gatives"
(self m<0) ifTrue: [self arrachement2]."[self dead:true]."
```

step:gens

```
"stocke dans les buffers les influences"
self calculMu.
self croissance:gens.!!
```

!Biofilm_Colonie methodsFor: 'id'!

```
CurrentId
    ^CurrentId isNil
        ifTrue: [0]
        ifFalse: [CurrentId!]

CurrentId: x
    ^CurrentId := x!

initId
    self id: (self CurrentId: self CurrentId + 1)!

!Biofilm_Colonie methodsFor: 'accessing'!

age
    ^age!

age: anObject
    age := anObject!

buffm
    ^buffm!

buffm: anObject
    buffm := anObject!

death
    ^death!

death: anObject
    death := anObject!

decay
    ^decay!

decay: anObject
    decay := anObject!

dynastie
    ^dynastie!

dynastie: anObject
    dynastie := anObject!

m
    ^m!

m: anObject
    m := anObject!

mu
    ^mu!

mu: anObject
    mu := anObject!

muM
    ^muM!

muM: anObject
    muM := anObject!
```

pere

^pere!

pere: anObject

pere := anObject!

rayon

^rayon!

rayon: anObject

rayon := anObject! !

!Biofilm_Colonie methodsFor: 'pov'!

povColonie

^#colonie!

povVivant

(self dead)

ifTrue: [#mort]

ifFalse: [#colonie].!!

Biofilm_Colonie subclass: #Biofilm_Aceto

```
instanceVariableNames: "
classVariableNames: 'CurrentId '
poolDictionaries: "
category: 'Cormas-Biofilm'
```

!Biofilm_Aceto methodsFor: 'init'!**init**

```
super init.
self m: "((Cormas random-0.5)/0.5*2+1)*"X10.    "+/- 20%"
self muM: Mu1Max*"((Cormas random-0.5)/0.5+1)".    "+/- 10%!"!
```

!Biofilm_Aceto methodsFor: 'procedure'!**calculMu**

```
"Calcule les taux spécifiques de chaque population bactérienne "
| ncol nlig |
self mu: self muM * self patch s1/ (self patch s1 + Ks1).
```

```
"Si taux spécifiques faibles, compteur de sous nutrition de la colonie"
```

```
self mu < (self muM / KFeed)
  ifTrue: [self death: self death + Dt]"
  ifFalse: [self death: 0].
```

```
"Vieillesse métabolique"
```

```
self age: ((self age) + Dt).
```

```
"disparition de biomasse active par arrachement, minéralisation,..."
```

```
nlig:= (((self patch id)-1) quo: Col)+1."num de la ligne "
ncol:= Col-(((self patch id) - ((nlig-1)*Col)))+1. "rem: Col)+1."
```

```
self decay: 2*Gamma/(TSH*TSH)*(ncol*DI)/(Thickness).
```

```
self death > 1.8 ifTrue: [self dead: true. self arrachement].!
```

croissance

```
super croissance.!
```

division: cellule dans: lesColonies vers: liste

```
"division cellulaire"
```

```
| colonie |
cellule effectif < EffSup ifTrue:
  [colonie := Biofilm_Aceto new.
  colonie init.
  colonie m: 0.
  colonie pere: self.
  self dynastie add: colonie.
  colonie moveTo: cellule.
  lesColonies add: colonie.
  liste add: colonie]!
```

reaction:repet

```
"Reaction chimique acétogène"
```

```
self patch buffs1: self patch buffs1 + (k11 * self mu * self m / (0.001*DI*DI*Df) * Dt / repet).
self patch buffs2: self patch buffs2+ (k12 * self mu * self m / (0.001*DI*DI*Df) * Dt / repet).! !
```

!Biofilm_Aceto methodsFor: 'control'!**step**

```
"to be completed"!!
```

!Biofilm_Aceto methodsFor: 'id'!

```
CurrentId
```

```
^CurrentId isNil
  ifTrue: [0]
  ifFalse: [CurrentId]!
```

```
CurrentId: x
```

```
^CurrentId := x!
```

```
initId
```

self id: (self CurrentId: self CurrentId + 1)!

!Biofilm_Aceto methodsFor: 'pov'!

povAceto
^#aceto!!

Biofilm_Colonie subclass: #Biofilm_Methano

```
instanceVariableNames: ""
classVariableNames: 'CurrentId '
poolDictionaries: ""
category: 'Cormas-Biofilm'
```

!Biofilm_Methano methodsFor: 'init'!**init**

```
super init.
self m: "(Cormas random)*X20.
self muM: Mu2Max*((Cormas random-0.5)/0.5+1)".!!
```

```
!Biofilm_Methano methodsFor: 'procedure'!
```

calculMu

```
"Calcule les taux spécifiques de chaque population bactérienne "
| ncol nlig |
self mu: self muM * self patch s2/ (self patch s2 + Ks2 + (self patch s2*self patch s2)/Ki2).
```

```
"Si taux spécifiques faibles, compteur de sous nutrition de la colonie"
```

```
self mu < (self muM / KFeed)
"
ifTrue: [self death: self death + Dt]"
ifFalse: [self death: 0].
```

```
"Vieillesse métabolique"
```

```
self age: ((self age) + Dt).
```

```
"disparition de biomasse active par arrachement, minéralisation,..."
```

```
nlig:= (((self patch id)-1) quo: Col)+1."num de la ligne "
ncol:= Col-(((self patch id) - ((nlig-1)*Col)))+1. "rem: Col)+1."
```

```
"nlig:= ((self patch id) quo: Col)-1.
ncol:= (((self patch id) - (nlig*Col)) rem: Col)+1."
self decay:2*Gamma/(TSH*TSH)*(ncol*DI)/(Thickness).!
```

croissance

```
super croissance.!
```

division: cellule dans: lesColonies vers: liste

```
"division cellulaire"
```

```
| colonie |
cellule effectif < EffSup ifTrue:
    [colonie := Biofilm_Methano new.
    colonie init.
    colonie m: 0.
    colonie pere: self.
    self dynastie add: colonie.
    colonie moveTo: cellule.
    lesColonies add: colonie.
    liste add: colonie]!
```

reaction:repet

```
"Reaction chimique methanogène"
```

```
self patch buffs2: self patch buffs2+ (k21 * self mu * self m / (0.001*DI*DI*Df) * Dt / repet).
self patch s3: self patch s3 + (k22 * self mu * self m / (0.001*DI*DI*Df) * Dt / repet)!!
```

!Biofilm_Methano methodsFor: 'control'!**step**

```
"to be completed"!!
```

!Biofilm_Methano methodsFor: 'id'!**CurrentId**

```
^CurrentId isNil
ifTrue: [0]
ifFalse: [CurrentId]!
```

CurrentId: x

```
^CurrentId := x!
```

initId

self id: (self CurrentId: self CurrentId + 1) !

!Biofilm_Methano methodsFor: 'pov'!

povMethano
^#methano! !

Modeling of Biofilms in Wastewater Treatment Processes using Multi-Agent Systems

The industrial and medical fundamental role of biofilms has led to many studies about their structures and their dynamic evolutions. This resulted in descriptive knowledge (*e.g.*, evolution of properties of the biofilm in space and time) and functional knowledge (*e.g.*, cell-cell communication) of biofilms. Many studies were thus devoted to build mathematical models for biofilm analysis but most of them were based on classical approaches (*e.g.*, using mass balance and space distribution in partial derivative equations). On the other side, only very few studies were directly based on the functional mechanisms to obtain virtual systems which mimic the descriptive observations and allow to predict the state of a biofilm and its possible evolution. In addition, in classical approaches, it is usually difficult to add new mechanisms in already developed models. This paper will present a non-classical approach based multi-agent systems for the analysis of biofilm formation in the anaerobic reactors which counteracts the above mentioned drawbacks. In addition, this model shows spatial repartition of biomass in the biofilm and the impact of the biofilm activity on its environment. The biomass is modeled by agents who are representing micro-colonies of identical bacteria. Each agent follows a mass balance law while its growth is described by a Monod law and its detachment is calculated as a function of its depth in the biofilm and of the level of hydrodynamic forces. The agents interact then with each other to find equilibrium between density or spatial growth. The diffusion-reaction problem has been solved in two dimensions by the Alternating Directions Implicit method. Several simulations studies will be proposed to account for characteristics of the influent liquid phase, for the diffusion of molecules in the biofilm and for the biochemical reaction.