

UNIVERSITÉ DE MONTRÉAL

Ce mémoire intitulé:

UTILISATION DE L'APPROCHE STOCHASTIQUE POUR LA SIMULATION DES
POLYMÉRISATIONS PAR OUVERTURE DE CYCLE, CAS DES POLYLACTIDES

OMAR FOUTLANE
DÉPARTEMENT DE GÉNIE CHIMIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CHIMIQUE)

Août 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-16781-6

Our file Notre référence

ISBN: 978-0-494-16781-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

UTILISATION DE L'APPROCHE STOCHASTIQUE POUR LA SIMULATION DES
POLYMÉRISATIONS PAR OUVERTURE DE CYCLE, CAS DES POLYLACTIDES

présenté par: FOUTLANE Omar

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. BERTRAND François, Ph.D., Président

M. DUBOIS Charles, Ph.D., directeur de recherche

Mme HEUZEY Marie-Claude, Ph.D., co directrice de recherche

M. DUFOUR Steven, Ph.D., membre

DÉDICACE

À ma famille et timouzgha.

REMERCIEMENTS

Je désire remercier mon directeur de recherche, le professeur Charles Dubois, et ma co directrice, la professeure Marie-Claude Heuzey, pour leurs conseils et leurs commentaires.

Je désire ensuite remercier tous les membres du groupe de recherche sur les polylactides, pour leur aide et leurs critiques notamment la professeure Paula Adams.

Je désire également remercier ma famille et mes amis pour leurs encouragements et leur soutien.

RÉSUMÉ

L'utilisation croissante de la modélisation et de la simulation stochastique dans la recherche scientifique a suscité un vif intérêt quant à son application dans la chimie des polymères. Ce travail s'inspire de travaux contemporains sur la simulation stochastique des systèmes linéaires pour traiter celle de la polymérisation par ouverture de cycle.

Ce mémoire a pour but principal de concevoir un code de calcul permettant par le biais de la simulation stochastique de :

- prédire l'évolution de la synthèse de polymères par ouverture de cycle connaissant les coefficients stochastiques régissant ce système
- estimer les vitesses de réactions principales se produisant dans le système réactionnel connaissant des résultats expérimentaux comme la conversion

Pour y parvenir nous avons élaboré un modèle probabiliste qui décrit globalement les événements principaux intervenant lors de la polymérisation par ouverture de cycle. Par la suite nous avons élaboré une approche pour estimer les coefficients stochastiques. Cette approche est basée sur le niveau de concordance des courbes estimées et expérimentales de l'évolution de paramètres mesurables de la polymérisation comme la conversion ou la masse molaire. Une fois les coefficients stochastiques estimés, il est possible d'en déduire les constantes de vitesse globales des réactions respectives.

Après avoir spécifié notre méthodologie, nous avons élaboré l'analyse et la conception des composantes de notre application que nous avons implémenté en utilisant le langage Fortran 95 selon le compilateur d'Absoft version 8.2 pour windows.

Enfin dans l'objectif de tester et valider notre application, nous avons choisi d'estimer les constantes de vitesse des réactions d'initiation, de propagation et de terminaison occurrentes lors de la synthèse des polylactides initiée par le benzyle

d'alcool, le 1,4-butanédiol ou le pentaérythritol. Pour cela nous avons utilisé les résultats expérimentaux issus de l'article de Korhonen et al. (2001). Les résultats que nous avons obtenus sont satisfaisants notamment pour l'alcool benzylique. Il s'avère de cette étude que le modèle adopté devra être amélioré afin d'obtenir de meilleurs estimations notamment dans le cas des alcools multifonctionnels. Des ajouts quant aux interactions utilisateur-application sont aussi préconisés dans le but de rendre notre travail plus attrayant.

ABSTRACT

The increasing use of stochastic modeling and simulation in scientific research brings a great interest for its application in polymer chemistry. This work takes as a starting point several contemporary contributions on the stochastic simulation of linear system to extent this approach to the case of the ring opening polymerization. The purpose of this thesis is to develop a computer code allowing, by the means of a stochastic simulation algorithm:

- The prediction the evolution of a ring opening polymerization system knowing the stochastic coefficients governing the reaction considered
- The estimation of apparent kinetics constants associated to the principal events occurring in a reactive system knowing some experimental results such as conversion or molecular weight.

In order to achieve this task we first elaborated a probabilistic model which describes the principal events found in typical ring opening polymerization. Thereafter we worked out an approach to best-estimate the stochastic coefficients. This estimation is carried out by a mathematical comparison of the simulated and experimental values of measurable parameters such as chemical conversion and molecular weight. Once the stochastic coefficients are estimated, it is possible to deduce the rate of respective reactions.

After having specified our methodology, we conducted the analysis and the design of the components of the resulting software, developed and written using the Absoft 8.2 Fortran 95 standard.

Finally, in order to test and validate our application, we choose to estimate the rates of the initiation, propagation and termination reactions occurring during the synthesis of the polylactides initiated by the benzyl alcohol, the 1,4-butanediol and the pentaerythritol . For that purpose we used the experimental results published by

of Korhonen et al. (2001). The results that we obtained are satisfactory in particular for the alcohol Benzyl. This study proves that the adopted model should be improved in order to obtain better estimates especially in the case of multifunctional alcohols. Additions are also recommended with an aim of making the interactions user-application simple and more attractive.

TABLE DES MATIÈRES

DÉDICACE	IV
REMERCIEMENTS	V
RÉSUMÉ	VI
ABSTRACT	VIII
TABLE DES MATIÈRES	X
LISTE DES TABLEAUX.....	XIII
LISTE DES FIGURES.....	XIV
LISTE DES ANNEXES.....	XVII
LISTE DES SYMBOLES ET ABRÉVIATIONS	XVIII
 CHAPITRE I : INTRODUCTION ET OBJECTIF	 1
 CHAPITRE II : REVUE DE LITTÉRATURE.....	 5
2.1 Notions Générales de la chimie des polymères.....	5
2.2 Méthodes de synthèse des polymères	8
2.2.1 Polycondensation	9
2.2.2 Polymérisation radicalaire.....	9
2.2.3 Polymérisation ionique.....	9
2.2.4 Polymérisation par ouverture de cycle.....	10
2.2.5 Réaction de transfert de chaîne	10
2.3 Polymérisation des lactides	11
2.3.1 Mécanismes réactionnels de la ROP des lactides	11
2.3.2 Polymérisation par ouverture de cycle par mécanisme cationique	12
2.3.3 Polymérisation par ouverture de cycle par mécanisme anionique	12

2.3.4 Polymérisation par ouverture de cycle par mécanisme de coordination-insertion	13
2.3.5 Catalyseurs et initiateurs	13
2.3.6 Réactions secondaires de transesterification	15
2.4 Théorie des graphes et application dans la reconnaissance des entités chimiques	15
2.4.1 Définition de graphe	16
2.4.2 Isomorphisme entre deux graphes	16
2.4.3 Méthodes de recherche d'isomorphismes de graphes	17
2.4.4 Application à la chimie des macromolécules	18
2.5 Processus stochastiques markoviens	19
2.5.1 Définition d'espace de probabilité ..	19
2.5.2 Définition de processus stochastique	20
2.5.3 Définition et caractéristiques des chaîne de Markov	21
2.5.4 Définition de processus de Markov	21
2.6 Modélisation des systèmes réactionnels chimiques	21
2.6.1 Approche déterministe ..	22
2.6.2 Approche statistique.....	22
2.6.3 Approche stochastique	23
2.6.4 Méthode de Gillespie	27
2.6.5 Méthode de Gibson	27
2.6.6 Méthode de Pandit.....	29
CHAPITRE III : MÉTHODOLOGIE	32
3.1 Objectif	32
3.2 Modélisation de la polymérisation des lactides	33
3.3 Estimation des constantes de réactions	37
3.4 Représentation du système réactionnel	38
3.5 Déroulement d'une simulation	41

CHAPITRE IV : ÉLABORATION DE L'APPLICATION ET RÉSULTATS DE L'ESTIMATION DE LA SYNTHÈSE DES POLYLACTIDE	46
4.1 Analyse fonctionnelle des besoins	46
4.1.1 Modèle ERD	47
4.1.2 Les diagrammes de flux des données.....	48
4.2 Algorithmes et structures de l'information	53
4.2.1 Structure des données	53
4.2.2 Algorithmes.....	54
4.3 Conception de l'application	58
4.4 Application à la synthèse des polylactides.....	60
4.4.1 Alcool benzylique	63
4.4.2 1,4-Butanédiol.....	68
4.4.3 Pentaérythritol.....	73
4.5 Interprétation.....	78
CHAPITRE III : DISCUSSION GÉNÉRALE	81
5.1 Synthèse des travaux	81
5.2 Discussion des résultats.....	82
5.3 Limitations des travaux	84
5.4 Conclusion	85
CHAPITRE VI : CONCLUSION GÉNÉRALE ET RECOMMANDATIONS.....	87
6.1 Conclusion générale	87
6.2 Perspectives de développements.....	88
BIBLIOGRAPHIE	90
ANNEXES	95

LISTE DES TABLEAUX

Tableau 2.1 Similitudes entre un graphe et une représentation moléculaire.....	18
Tableau 4.1 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence de l'alcool benzylique avec 1 source	63
Tableau 4.2 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence de l'alcool benzylique avec deux sources.....	63
Tableau 4.3 Valeurs des constantes de réactions de la synthèse des polylactides en présence de l'alcool benzylique	66
Tableau 4.4 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du 1,4-butanédiol avec 1 source.....	68
Tableau 4.5 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du 1,4-butanédiol avec 2 sources...	68
Tableau 4.6 Valeurs des constantes de réactions de la synthèse des polylactides en présence du 1,4-butanédiol.....	71
Tableau 4.7 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du pentaérythritol avec 1 source	73
Tableau 4.8 Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du pentaérythritol avec 2 sources...	73
Tableau 4.9 Valeurs des constantes de réactions de la synthèse des polylactides co- initié par le pentaérythritol	76

LISTE DES FIGURES

Figure 2.1 Allure de la distribution des masses molaires d'un polymère.	7
Figure 2.2 Structures de polymères ramifiés.....	8
Figure 2.3 Structure de polymères réticulés.....	8
Figure 2.4 Schéma de développement de chaînes par polycondensation	9
Figure 2.5 Polymérisation par ouverture de cycle cationique.....	12
Figure 2.6 Polymérisation par ouverture de cycle anionique.....	12
Figure 2.7 Polymérisation par ouverture de cycle par coordination-insertion.....	13
Figure 2.8 Mécanisme de Kricheldorf.	14
Figure 2.9 Mécanisme de Penczeck.	14
Figure 2.10 Réactions de transesterification intermoléculaire.....	15
Figure 2.11 Réactions de transesterification intramoléculaire.....	15
Figure 2.12 Exemple de deux graphes isomorphes.....	16
Figure 2.13 Schéma représentant l'évolution du système réactionnel.....	24
Figure 2.14 Exemple de graphe de dépendance de la méthode de Gibson.....	28
Figure 3.1 Initiation de la polymérisation des lactides.	33
Figure 3.2 Schéma de propagation d'une chaîne de polylactide co-initiée par un trialcool	62
Figure 3.3 Molécule de polylactide initiée par un glycolide.....	39
Figure 3.4 Modèle d'une chaîne.....	40
Figure 4.1 Diagramme entité association.....	47
Figure 4.2 Diagramme de contexte.	48
Figure 4.3 DFD de niveau 1.....	49
Figure 4.4 DFD de niveau 2 de la simulation	50
Figure 4.5 DFD de niveau 3 de la simulation correspondant à ordonner les chaînes.....	51
Figure 4.6 DFD de niveau 2 d' « Afficher graphiques ».	51
Figure 4.7 DFD de niveau 2 d' « Écrire résultats »	52
Figure 4.8 DFD de niveau 2 d' « Estimer constantes de réaction ».....	52

Figure 4.9 Diagramme de conception des modules de l'application.	59
Figure 4.10 Estimation de la distribution massique M_n dans le cas de l'alcool benzylique après 5, 50 et 100 simulations respectivement ($I_0=500$)	62
Figure 4.11.a Comparaison de la conversion expérimentale et estimée pour la synthèse des polylactides co-initiée par l'alcool benzylique.	65
Figure 4.11.b Comparaison de M_w expérimental et estimé pour la synthèse des polylactides co-initiée par l'alcool benzylique.	65
Figure 4.12.a Estimation de la distribution massique lors de la synthèse des polylactides co-initiée par l'alcool benzylique.	67
Figure 4.12.b Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par l'alcool benzylique	67
Figure 4.13.a Comparaison de la conversion expérimentale et estimée pour la synthèse des polylactides co-initiée par le 1,4-butanédiol.	70
Figure 4.13.b Comparaison de M_w expérimental et estimé pour la synthèse des polylactides co-initiée par le 1,4-butanédiol.	70
Figure 4.14.a Estimation de la distribution massique lors de la synthèse des polylactides co-initiée par le 1,4-butanédiol.	72
Figure 4.14.b Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par le 1,4-butanédiol.	72
Figure 4.15.a Comparaison de la conversion expérimentale et estimée pour la synthèse des polylactides co-initiée par le pentaérythritol.....	75
Figure 4.15.b Comparaison du M_w expérimental et estimé pour la synthèse des polylactides co-initiée par le pentaérythritol.....	75
Figure 4.16.a Estimation de la distribution massique lors de la synthèse des polylactides co-initiée par le pentaérythritol.....	77
Figure 4.16.b Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par le pentaérythritol.....	77
Figure 4.17.a Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par le pentaérythritol ($k_t=2 \text{ e-}2$).	80

Figure 4.17.b Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par le pentaérythritol ($k_i=3 \text{ e-3}$).	80
---	----

LISTE DES ANNEXES

Annexe I fichier entrees.txt	95
Annexe II Exemple du fichier « optimisation.txt »	96
Annexe III Extrait du fichier « polyfinal.txt »	98
Annexe IV Extrait du fichier « proba.txt »	100
Annexe V Extrait du fichier « moments.txt »	103
Annexe VI Code du programme	106

LISTE DES SYMBOLES ET ABRÉVIATIONS

$A :$	événement
$A, B \text{ et } C :$	symboles de réactifs ou produits chimiques
$a, b :$	<i>nombre de molécules des espèces A B</i>
$A^c :$	événement complémentaire
$A_S A_S' :$	désigne l'ensemble
$a_\mu a_\alpha :$	fonction intensité de probabilité
$c=(c_b, c_p, c_t) :$	vecteur des coefficients stochastiques
$c_l, c_i, c_p, c_t :$	coefficients stochastiques
$CV :$	la conversion
$dc_b, dc_p, dc_t :$	pas de recherche des coefficients stochastiques
$DFD :$	décomposition des flux de données
$DP :$	degré de polymérisation
$DPI :$	degré de polydispersité
$E E^j :$	états du système réactionnel
$E(Y_i) :$	espérance de Y_i
$EG, EG_o \text{ et } EG' :$	états de graphes
$e_i :$	j ième événement
$ERD :$	entité relation association
$ES_1, ES_2, \dots, ES_i :$	espèces chimiques
$E_X :$	espace d'états pour un processus X
$f, ff_1, ff_2, ff_n :$	fonctions
$F :$	une tribu
$F(EG, n, m) :$	fonction d'affinement de l'algorithme de Cordella
$f_I :$	fonction objective avec une source

$f_2 :$	fonction objective avec deux sources
$f_c(Y,I) :$	fonction de paramètre c et d'arguments Y et I
$f_e :$	facteur d'échelle de Pandit
$G, G', G_S, G_{S'} :$	graphes
$g_1, g_2 :$	groupes fonctionnels
$h_\mu :$	nombre de combinaisons de réactifs de la réaction R_μ
$I(t) :$	données des sources d'information
$I_o :$	nombre initial de co-initiateur
$k_i, k_p, k_t :$	constantes de vitesse de réactions
$L :$	nombre de sources d'information
$M = (a_{ij}) :$	matrice d'adjacence
$Mc :$	masse molaire de la chaîne
$M(EG) :$	ensemble des isomorphismes entre deux graphes
$M_2 :$	normalisateur du moment M_w
$M_i :$	masse molaire de l'espèce i
$ML :$	matrice des liaisons
$M_n :$	premier moment de la distribution massique
$M_o :$	nombre initial des monomères
$M_u :$	masse molaire de l'unité répétitif
$M_w :$	deuxième moment de la distribution massique
$M_z :$	troisième moment de la distribution massique
$N :$	le nombre de points observés avant l'arrêt de la synthèse
$n, m, d, n', m' \text{ et } d' :$	longueur de branches de chaîne
$N_\theta :$	nombre initial de sites non initialisés
$N_a :$	nombre d'Avogadro
$N_i :$	nombre de l'espèce i
$N_S, N_{S'} :$	nombre de sommets du graphe G_S
$p :$	degré de polymérisation de chaque branche

$P :$	mesure de probabilité
$p(x; t) :$	probabilité d'avoir une population moléculaire x à t
$p^* :$	probabilité de repos du système
$p_1, p_2, p_3, p_4 :$	probabilités des événements
$PP(EG), :$	ensemble des paires de sommets candidates
$PV :$	vecteur de probabilités
$r :$	variable aléatoire exponentielle
$R(OH)_x, R-(OH)_{x-1} :$	représentation d'alcools
$R_j, R_\mu, R_\alpha :$	réactions chimiques
$S, S' :$	ensemble des sommets d'un graphe
$s_i s_j :$	sommet d'un graphe
$T :$	ensemble des indices t
$t^* :$	temps de repos du système
tab:	tableau des chaînes dans le système à l'instant t
$Tf :$	temps final d'arrêt de la réaction
$URN :$	valeur aléatoire unitaire
$V :$	volume réactionnel
$\Omega :$	ensemble de résultats d'une expérience aléatoire
$X = (X_1, X_2, X_3, \dots, X_n) :$	vecteur de particules des espèces dans un système
$X_1(t) :$	nombre de monomères à l'instant t
$X_2(t) :$	nombre de sites réactionnels actifs à l'instant t
$X_3(t) :$	nombre de sites initialisés jusqu'à l'instant t
$X_i :$	nombre de particule de l'espèce i
$X_t :$	processus stochastique
$Y(t) :$	données estimées par le modèle
$\alpha_{hj}(t) :$	probabilité de transition de l'état h à j
$\alpha_l :$	poids de la source l
$--OH^* :$	un site réactionnel actif
$--OH :$	un site réactionnel non initialisé

$--OH_T:$	un site réactionnel qui a subi une terminaison
$--M--OH:$	une chaîne
$\Delta t :$	intervalle de temps
$\mathbf{v}_\mu :$	vecteur de transition stoechiométrique de $\mathbf{R}_\mu$
$\tau_\mu:$	temps d'attente
$\sigma_n :$	écart type de la distribution massique

CHAPITRE I

INTRODUCTION ET OBJECTIFS

La modélisation et la simulation stochastique des phénomènes chimiques présentent de nombreux défis vu la complexité des processus et le nombre de paramètres qui y interviennent. Toutefois, il faut garder à l'esprit que la modélisation et la simulation de phénomènes chimiques ne prétendent pas reproduire ceux-ci exactement mais plutôt à proposer des méthodes qui soient capables de fournir un certain nombre de résultats approximant au mieux certaines des caractéristiques du phénomène réel. Ainsi, la simulation est un outil qui sert de complément d'aide à la synthèse et au contrôle des systèmes chimiques. Elle sert à prédire l'évolution des systèmes chimiques et par la suite à mieux contrôler les paramètres influençant cette évolution tout en minimisant les coûts induits par une multitude d'expériences.

Dans le passé, les méthodes de simulation étaient peu utilisées en raison de la nécessité d'utiliser des machines de calcul puissantes et des modèles adéquats. Ce constat est totalement changé avec l'avènement des ordinateurs et les progrès réalisés dans le domaine de la technologie de l'information, tant sur le plan du matériel que des logiciels d'une part, et par les avancées mathématiques dans la théorie probabiliste d'autre part. Aussi, de plus en plus de scientifiques ont proposé des modèles stochastiques pour décrire les phénomènes chimiques et un nombre grandissant d'articles traitant le sujet de la simulation stochastique des phénomènes chimiques apparaissent dans les revues scientifiques spécialisées. Cette tendance trouve son explication dans le fait que l'approche stochastique permet une modélisation très proche de la réalité des phénomènes étudiés. En effet, de façon générale, l'approche stochastique se propose de reproduire les caractéristiques macroscopiques d'un système à partir du comportement microscopique de chacun des objets le constituant. L'observation de l'ensemble du système se faisant à l'échelle macroscopique, un grand

nombre d'objets seront simulés de façon à rendre les résultats statistiquement significatifs. Dans le cas des réactions chimiques, les objets sont des molécules et les phénomènes sont donc illustrés de façon dynamique et à l'échelle microscopique.

Plusieurs chercheurs se sont intéressés à cette approche de la modélisation et de la simulation des réactions chimiques, parmi lesquels on trouve :

- Gillespie (1977)
- Van Kampen et al. (1992)
- Pandit et al. (1993)

Dans ce travail nous allons proposer un modèle probabiliste pour la polymérisation par ouverture de cycle des lactides (ou « Ring Opening Polymerization », ROP), et nous allons nous servir des simulations stochastiques pour déterminer les paramètres de ce modèle appliqué à la ROP des lactides initiée par différents alcools en présence du co-initiateur octoate d'étain. Les paramètres du modèle seront déterminés par régression linéaire basée sur le critère de minimisation des moindres carrés. Lors de cette étude nous allons utiliser des résultats de l'article de Korhonen et al. (2001). Pour ce faire nous avons élaboré un programme de simulation de la ROP des lactides avec des alcools de fonctionnalités différentes. Cette approche pour décrire la ROP des lactides pourra être facilement adaptée pour décrire d'autres réactions chimiques moyennant des transformations minimales du programme de simulation élaboré. Le choix de la ROP des lactides n'est pas restrictif. Il a été motivé par le fait de l'intérêt que suscite actuellement les polymères à base d'acide lactique. Ces polymères préparés en présence d'octoate d'étain sont biodégradables, proviennent de sources renouvelables et ne présentent aucune toxicité pour l'organisme (Gilding and Reed, (1979)). Ils peuvent dès lors être utilisés dans tous les domaines de la vie courante. Notre travail vise à fournir un moyen permettant le contrôle des paramètres importants de la nature des polymères comme la composition, la longueur de chaînes, l'indice de polydispersité, etc. Le reste de ce travail est structuré de la manière suivante:

- Le deuxième chapitre met l'accent sur les aspects et les résultats théoriques qui ont été nécessaires pour mener à bien cette étude, ainsi que les résultats publiés jusqu'à présent et qui touchent de près les objectifs de ce travail, notamment la simulation stochastique des réactions chimiques qui reste le noyau central de ce projet. Un accent important est mis sur la description des réactions de polymérisation des lactides en particulier. Des rappels théoriques sur les processus markoviens nécessaires à la modélisation stochastique des réactions chimiques y sont présentés, de même que des rappels de la théorie des graphes et son application à la chimie, notamment à la reconnaissance des entités chimiques. Le chapitre se termine par une explication des principales méthodes de simulation stochastique adoptées jusqu'à nos jours dans le domaine de la chimie en général et de la chimie des polymères en particulier.
- Le troisième chapitre expose la méthodologie que nous avons suivie pour proposer notre modèle pour les réactions par ouverture de cycle des lactides. Nous y présentons aussi le principe de détermination des paramètres du modèle qui repose sur une minimisation d'une fonction objective que nous avons définie.
- Le quatrième chapitre présente les étapes d'élaboration du programme de simulation stochastique et discute les résultats obtenus pour la ROP des lactides initiée par différents alcools et offre des explications et interprétations de ces résultats. Il est question de traiter les caractéristiques de la relation liant le modèle et le phénomène modélisé. À cette fin, nous allons nous servir des résultats expérimentaux rapportés dans l'article de Korhonen et al. (2001).
- Le cinquième chapitre présente une discussion générale sur les résultats et la méthodologie que nous avons utilisée pour étudier la polymérisation par ouverture de cycle des lactides. Il propose des solutions aux problèmes que nous

avons rencontrés, notamment lors de la comparaison des résultats du modèle avec ceux des expériences rapportées dans l'article de Korhonen et al. (2001).

- Une conclusion et des recommandations en vue de travaux futurs ainsi que les applications possibles de cette étude sont présentées au sixième chapitre qui clôture ce mémoire. Nous y proposons des améliorations et des ajouts susceptibles de rendre l'approche adoptée plus performante.

CHAPITRE II : REVUE DE LITTÉRATURE

La modélisation et la simulation sont aujourd'hui très présentes dans les sciences et s'illustrent essentiellement dans la prédiction; C'est aussi une voie pour comprendre les systèmes complexes. Pour ce faire, il faut avoir compris les processus qui interviennent de près ou de loin dans l'évolution des systèmes modélisés. Aussi avons-nous jugé utile de rappeler les outils que nous avons utilisés lors de l'élaboration de ce projet, à savoir:

- les notions générales de la chimie des polymères;
- les principes de la polymérisation des lactides;
- les applications de la théorie des graphes dans la reconnaissance des entités chimiques;
- les principes des processus stochastiques markoviens;
- les approches de modélisation des systèmes réactionnels chimiques.

2.1 Notions générales de la chimie des polymères

Les polymères synthétiques présentent des propriétés qui leur permettent dans de nombreux cas, non seulement de remplacer les métaux, mais aussi de les surpasser. Ceci ne fait que promouvoir les recherches dans le domaine de la synthèse des polymères. Les définitions des plus importants concepts utilisés dans la synthèse des polymères sont:

- **Macromolécule:** molécule composée d'un grand nombre d'atomes ou de motifs moléculaires répétitifs. Le motif pourra être composé d'une ou plusieurs unités de base.

- **Polymère** : macromolécule constituée de la répétition d'un motif moléculaire appelé unité structurelle ou de répétition. On distingue les polymères linéaires, branchés, hyperbranchés, etc.
- **Monomère** : réactif utilisé comme unité répétitive pour l'obtention d'un polymère. Un homopolymère est un polymère synthétisé à partir d'un seul type de monomère, alors que le terme copolymère est utilisé lorsque plusieurs monomères différents sont présents.
- **Initiateur** : substance chimique qui en subissant une transformation chimique ou en présence d'autres espèces chimiques, est capable d'initier la polymérisation. Le choix de l'initiateur dépend du milieu dans lequel est effectuée la réaction de synthèse et des propriétés requises du produit polymère.
- **Degré de polymérisation (DP)** : longueur de la chaîne en fonction du nombre de fois que l'unité structurelle est répétée dans celle-ci.

$$DP = M_c / M_u \quad (2-1)$$

où M_c désigne la masse molaire de la chaîne et M_u désigne la masse molaire du motif répétitif.

- **Distribution des masses molaires** : lors de la synthèse de polymères, il est difficile de contrôler exactement le nombre de monomères qui s'ajoutent à une chaîne avant que celle-ci se termine. On obtient donc des chaînes de longueur différentes et on parle de distribution de masse plutôt que d'une valeur spécifique. Pour bien définir un échantillon, il faut posséder de l'information sur la largeur de la distribution des masses molaires. Le plus souvent, la masse molaire moyenne en poids et en nombre sont utilisées pour caractériser un polymère. On définit ces deux concepts comme suit :

$$M_n = \sum N_i M_i / \sum N_i \quad (2-2)$$

$$M_w = \sum N_i M_i^2 / \sum N_i M_i \quad (2-3)$$

où N_i : nombre de moles de molécules de masse molaire M_i dans l'échantillon

La distribution des masses molaires a habituellement, mais pas nécessairement, la forme d'une distribution normale. Les anomalies observées sont dues aux réactions secondaires et aux conditions régissant la polymérisation dans le réacteur. La figure 2.1 illustre cette distribution.

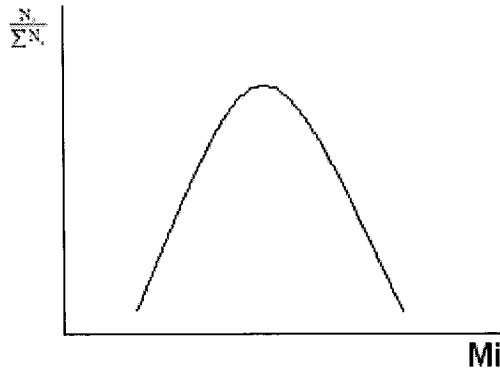


Figure 2.1: Allure de la distribution des masses molaires d'un polymère.

- **Indice de polydispersité (DPI) des polymères :** l'indice de polydispersité représente l'étendue de la distribution. Il est donc relié à l'écart type de la distribution des masses molaires (σ_n). On définit le *DPI* par le rapport entre la masse molaire moyenne en poids et la masse molaire moyenne en nombre:

$$DPI = M_w / M_n \quad (2-4)$$

- **Polymère linéaire :** dans un polymère à structure linéaire, les monomères sont placés les uns à la suite des autres. Ainsi, en étirant le polymère nous obtenons une allure de droite.
- **Polymères ramifiés :** un polymère ramifié peut être branché, hyperbranché ou greffé. On parle de polymère ramifié lorsque le même motif constitue les

branches et la chaîne principale, de polymère greffé lorsque les branches sont composées d'un autre motif et de polymère en étoile ou de dendrimères lorsque le polymère ramifié a une structure régulière de forme sphérique. La figure 2.2 présente ces trois différents types.

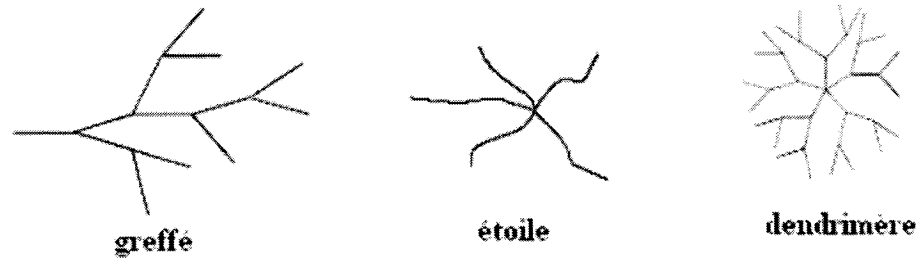


Figure 2.2 : Structures de polymères ramifiés.

- **Polymères réticulés** : Un polymère est dit réticulé si les chaînes de polymère sont liées de façon à former un réseau tridimensionnel. Cette architecture réticulée particulière donne lieu à d'immenses molécules. La figure 2.3 illustre cette structure.

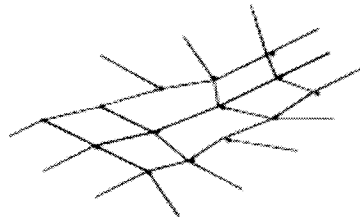


Figure 2.3 : Structure de polymères réticulés.

2.2 Méthodes de synthèse des polymères

L'obtention d'un polymère peut s'effectuer par plusieurs méthodes de synthèse différentes. Leur classification est basée sur une description du mécanisme de la croissance des chaînes. Nous citons parmi les plus utilisées en industrie la polycondensation, la polymérisation radicalaire, la polymérisation ionique et la polymérisation par ouverture de cycle.

2.2.1 Polycondensation

La synthèse de polymères par polycondensation est très utilisée dans l'industrie. La polycondensation des monomères bifonctionnels produit des chaînes linéaires tandis que la polycondensation des monomères multifonctionnels produit des chaînes réticulées et parfois hyperbranchées. Cette voie synthétique permet généralement de contrôler la structure moléculaire, la masse molaire et la polydispersité. Ce contrôle s'effectue en utilisant des composés monofonctionnels afin de diminuer les masses molaires ou en agissant sur le temps de la réaction, ce qui permet de contrôler la taille des polymères formés. Si on désigne par g_1 et g_2 les groupes fonctionnels des monomères ou les unités de répétition, un schéma simplifié du mécanisme de développement des chaînes peut être représenté comme illustré dans la figure 2.4:

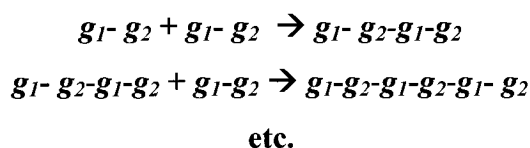


Figure 2.4 : Schéma de développement de chaînes par polycondensation.

Lorsque les chaînes du polymère se développent sans élimination de petites molécules, on parle généralement de polymérisation par addition ou radicalaire.

2.2.2 Polymérisation radicalaire

La polymérisation radicalaire est l'une des voies synthétiques les plus utilisées dans la synthèse des polymères. La polymérisation radicalaire est initiée par la formation de radicaux libres. Ces derniers vont stimuler les réactions de propagation et par la suite la formation de chaînes de polymère.

2.2.3 Polymérisation ionique

La polymérisation ionique est stimulée par un initiateur ionique. Selon la nature de cet initiateur, le polymère forme une espèce chargée anion ou cation. On distingue la

polymérisation anionique et la polymérisation cationique. Dans certains cas on peut arrêter la polymérisation en cessant de l'alimenter en monomère, puis la reprendre avec un monomère différent pour obtenir un copolymère séquencé, de telles pratiques permettent le contrôle des propriétés des polymères produits.

2.2.4 Polymérisation par ouverture de cycle

La polymérisation par ouverture de cycle est largement utilisée pour la synthèse de polyesters et de polyéthers. Depuis les travaux publiés de Carothers et al. (1932) beaucoup de chercheurs ont intensivement exploré la technique de la polymérisation par ouverture de cycle pour des lactones, des anhydrides, et des carbonates. La méthode a été appliquée à une diversité de monomères pour produire plusieurs types de polymères et un bon nombre d'initiateurs et de catalyseurs ont été développés. Généralement, cette voie synthétique utilise des mécanismes anionique, cationique ou par coordination-insertion.

2.2.5 Réaction de transfert de chaîne

Lorsque la chaîne en croissance réagit avec une autre espèce chimique que le monomère, on parle de transfert de chaîne. Cette réaction peut se produire avec le solvant, une impureté, un autre polymère, un autre monomère ou un agent de transfert de chaîne, ajouté volontairement pour diminuer la masse molaire.

Une polymérisation est dite «vivante» lorsque les processus de terminaison sont négligeables, voire inexistant dans le système réactionnel. On obtient alors des polymères de grande masse molaire.

2.3 Polymérisation des lactides

Parmi les polymères biodégradables, les polyesters aliphatiques, et plus particulièrement les polylactides, connaissent des développements industriels tels qu'ils apparaissent aujourd'hui comme des substituts réels aux polymères traditionnels issus de la pétrochimie. Généralement les polylactides sont obtenus par polymérisation

d'ouverture de cycle (ROP) des lactides, comme il est mentionné par Zhang (1998) et Nijenhuis (1992) . Ils sont utilisés dans de nombreux domaines, notamment comme les implants médicaux et les systèmes de délivrance de médicaments (Kricheldorf (2001), Okada (2002)).

Les polylactides peuvent être préparés par deux approches différentes, la polycondensation de l'acide lactique ou la polymérisation par ouverture de cycle (ROP) des lactides. La polymérisation par polycondensation est moins coûteuse que la polymérisation par ouverture de cycle. Toutefois les problèmes associés à la polycondensation, telle que le besoin de stoechiométrie exacte, les températures élevées de la réaction, la difficulté d'atteindre des masses molaires élevées et le besoin d'éliminer les sous-produits à faible poids moléculaire (par exemple l'eau) font que la tendance dans la production des polylactides est d'utiliser la polymérisation par ouverture de cycle.

2.3.1 Mécanismes réactionnels de la polymérisation par ouverture de cycle (ROP) des lactides

Les polylactides de poids moléculaire élevé sont exclusivement produits par ouverture de cycle (Uhrich et al. (1999)). La réaction de polymérisation par ouverture de cycle peut être conduite en masse ou en solution (Sosnowski et al. (1996)). Selon la nature de l'initiateur, la polymérisation procède selon trois principaux mécanismes réactionnels différents : le mécanisme cationique, le mécanisme anionique, ou le mécanisme de coordination-insertion (Johns et al. (1984)).

2.3.2 Polymérisation par ouverture de cycle par mécanisme cationique

La polymérisation cationique par ouverture de cycle implique la formation d'une espèce chargée qui réagira ensuite avec un monomère (Albertsson et Palmgren (1996)). La réaction a comme conséquence une ouverture du cycle de l'espèce chargée comme le montre la figure 2.5 :

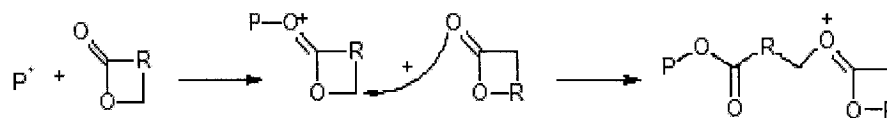


Figure 2.5 : Polymérisation par ouverture de cycle cationique.

La polymérisation cationique est difficile à contrôler et souvent seulement des polymères à faible poids moléculaire en sont issus (Penczek et al. (2000)).

2.3.3 Polymérisation par ouverture de cycle par mécanisme anionique

La polymérisation anionique par ouverture de cycle des lactides a lieu suite à la réaction d'un initiateur négativement chargé avec le carbone du groupement carboxylique, ce qui induit la scission de la liaison acyle-oxygène. L'espèce en propagation est négativement chargée et est équilibrée par un cation. Un exemple de ce mécanisme est présenté par la figure 2.6. Une des meilleures méthodes de contrôle menant aux polymères de haut poids moléculaire est la polymérisation anionique effectuée dans un solvant polaire.



Figure 2.6 : Polymérisation par ouverture de cycle anionique.

2.3.4 Polymérisation par ouverture de cycle par mécanisme de coordination-insertion

La polymérisation par ouverture de cycle par mécanisme de coordination-insertion est désignée souvent par le terme polymérisation pseudo-anionique par ouverture de cycle, puisque la propagation procède par la coordination du monomère à

l'espèce active, suivi de l'insertion du monomère dans la liaison R-oxygène. La figure 2.7 montre une présentation schématique du mécanisme de coordination-insertion.

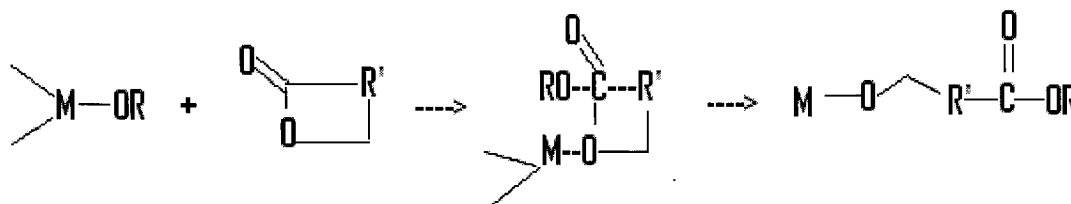


Figure 2.7 : Polymérisation par ouverture de cycle par coordination-insertion.

2.3.5 Catalyseurs et initiateurs

La polymérisation par ouverture de cycle des lactides est influencée par différents paramètres, parmi lesquels on peut citer la nature du catalyseur utilisé. Les principaux catalyseurs utilisés dans la ROP des lactides sont:

- **Octoate d'étain (II) :**

L'octoate d'étain $\text{Sn}(\text{Oct})_2$, est fréquemment utilisé dans la ROP des lactides. La polymérisation s'effectue par coordination-insertion, où un groupe fonctionnel d'hydroxyle va être coordonné au $\text{Sn}(\text{Oct})_2$ pour former un complexe initiateur. Les deux chemins réactionnels les plus connus sur le mécanisme réactionnel de coordination-insertion en présence d'octoate d'étain(II) sont : le premier mécanisme illustré dans la figure 2.8 est proposé par Kricheldorf, (Kricheldorf et al. (2000)), où le co-initiateur, le groupe hydroxyle, et le monomère sont coordonnés au $\text{Sn}(\text{Oct})_2$ lors de la propagation. L'autre mécanisme réactionnel illustré dans la figure 2.9 est proposé par Penczeck, (Penczeck et al. (1998)), où le complexe $\text{Sn}(\text{Oct})_2$ est converti en alkoxide d'étain avant de participer à la polymérisation:

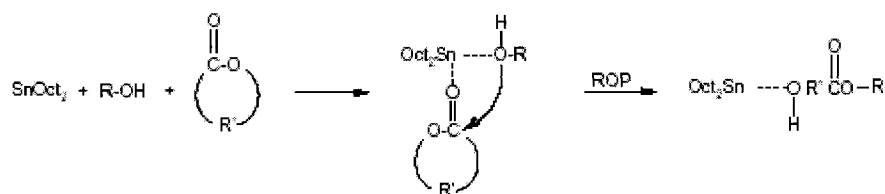


Figure 2.8 : Mécanisme de Kricheldorf.

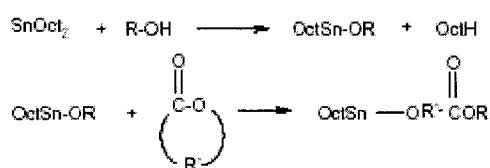


Figure 2.9 : Mécanisme de Penczeck.

- **Tri-isopropoxyde d'aluminium :**

La polymérisation par ouverture de cycle des lactides en utilisant le tri-isopropoxyde d'aluminium a été étudiée par beaucoup de chercheurs. Selon Quirk et Lee (1992) l'utilisation de cet initiateur permet d'obtenir des polymères à structure prédéfinie. Le mécanisme réactionnel est celui de la coordination-insertion où la ROP des lactides est caractérisée par la quasi-absence de réactions secondaires selon les travaux de recherche de Dubois (Dubois et al. (1991)).

- **Alkoxides d'étain (IV) :**

Les alkoxides d'étain (IV) et les alkoxides d'étain cycliques sont utilisés comme initiateurs de la polymérisation par ouverture de cycle des esters cycliques et des lactides. Le mécanisme proposé est celui de la coordination-insertion. Les travaux de recherche entrepris par Stridsberg (Stridsberg et al. (2000)) ont révélé que la masse molaire moyenne croît linéairement avec le taux de conversion et que la distribution des masses molaires est étroite, le *DPI* est de l'ordre de 1.15. Ceci est expliqué par la quasi absence des réactions de transfert.

2.3.6 Réactions secondaires de transestérification

Lors des réactions de polymérisation par ouverture de cycle des lactides, l'évolution du système réactionnel peut donner naissance à des réactions secondaires, les plus importantes sont celles de transestérification. Ces dernières peuvent être d'ordre intermoléculaire ou d'ordre intramoléculaire (« back-biting ») causant la dégradation de la chaîne principale du polymère obtenu et menant à la formation d'oligomères cycliques. Les réactions de transestérification ont pour effet d'élargir la distribution des masses molaires des polylactides. La figure 2.10 illustre les réactions de transestérification intermoléculaire, quant à la figure 2.11, elle montre les réactions de transestérification intramoléculaires.

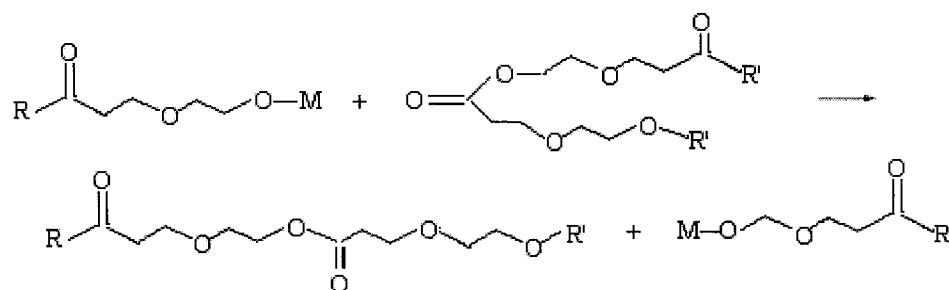


Figure 2.10 : Réactions de transestérification intermoléculaire.

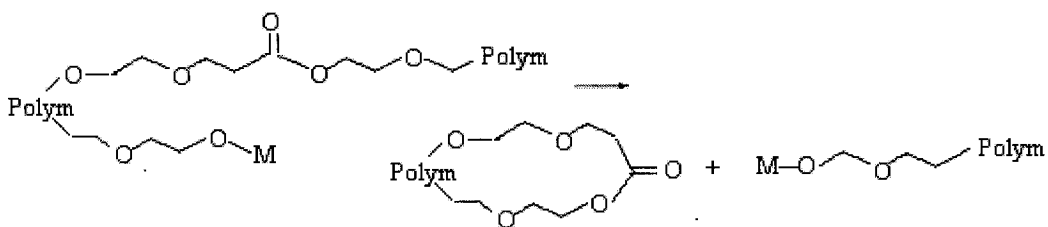


Figure 2.11 : Réactions de transestérification intramoléculaire.

2.4 Théorie des graphes et application dans la reconnaissance des entités chimiques :

La théorie des graphes permet de modéliser certains phénomènes pour ainsi aboutir à des solutions efficaces pour de nombreux problèmes. Elle est présente dans plusieurs applications et intervient dans des domaines variés comme la chimie, la

logistique, les mathématiques, etc. La théorie des graphes est un vaste champ qui contient plusieurs résultats intéressants. Dans notre cas, nous l'utilisons pour reconnaître les macromolécules identiques contenues dans le système réactionnel, ce qui nous permet par la suite de déterminer la distribution des différentes espèces chimiques du système étudié.

2.4.1 Définition de graphe

Un graphe G orienté est défini par le couple $G_S = (S, A_S)$, où S désigne l'ensemble des sommets de G_S , et A_S est l'ensemble des arcs de G_S . s_j est successeur de s_i si $(s_i, s_j) \in A_S$; L'ensemble des successeurs de s_i est noté $Succ(s_i)$ s_i est prédécesseur de s_j si $(s_i, s_j) \in A_S$; L'ensemble des prédécesseurs de s_i est noté $Pred(s_i)$. Il y a plusieurs façons de représenter un graphe, la plus simple étant celle de la matrice adjacente. La matrice adjacente associée à un graphe à n sommets $s_1, s_2, \dots, s_n$ est la matrice carrée $M = (a_{ij})$, tel que:

$$\begin{aligned} a_{ij} &= 1 \text{ si il y a un arc de } s_i \text{ vers } s_j \\ a_{ij} &= 0 \text{ sinon.} \end{aligned} \quad (2-5)$$

2.4.2 Isomorphisme entre deux graphes

Le problème de l'isomorphisme est un problème fondamental en théorie des graphes. On peut le formuler de la façon suivante : soient deux graphes $G_S = (S, A_S)$ et $G_{S'} = (S', A_{S'})$. G_S et $G_{S'}$ sont *isomorphes* s'il existe une bijection $f: S \rightarrow S'$ telle que :

$$\forall s_i, s_j \in S : (s_i, s_j) \in A_S \Leftrightarrow (f(s_i), f(s_j)) \in A_{S'}$$

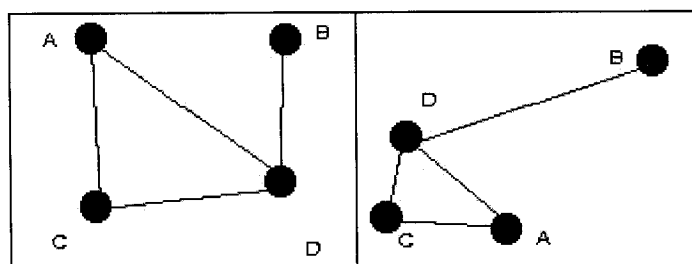


Figure 2.12 : Exemple de deux graphes isomorphes.

2.4.3 Méthodes de recherche d'isomorphismes de graphes

On rencontre le problème de l'isomorphisme de graphes dans de nombreux domaines parmi lesquels la chimie (Rouvray et Balaban, (1979)). En effet, le problème de la reconnaissance des molécules qui possèdent des structures identiques, appelées isomères, est apparenté à celui de l'isomorphisme des graphes. Il y a plusieurs méthodes pour résoudre le problème de l'isomorphisme de graphes, dans notre étude nous avons choisi l'algorithme cité dans l'article de Cordella et al. (2004) qui est une amélioration de l'algorithme d'Ullman (1976) : On construit un tableau de dimension $N_S \times N_{S'}$, où N_S (resp. $N_{S'}$) désigne le nombre de sommets du graphe G_S (resp. $G_{S'}$), en exécutant une procédure récursive admettant ce tableau comme paramètre qui est modifié à chaque itération par une méthode de raffinement.

Présentation de l'algorithme de Cordella et al. :

L'algorithme se base sur la procédure « match » dont le but est de chercher récursivement l'existence d'un isomorphisme entre deux graphes G et G' . En partant de l'état initial EG_o vide, les paires de nœuds similaires dans G et G' sont ajoutées au fur et à mesure que l'on prospecte les graphes en question.

Procédure match (EG)

Entrée : un état EG constitué des couples (s_i, s_j) isomorphes

Sortie : ensemble des isomorphismes $M(EG)$ entre deux graphes

Si $\text{card } M(EG) = \text{card } (G')$ alors

afficher $M(EG)$

Sinon

Identifier $PP(EG)$, ensemble des paires de sommets candidates à être incluses dans $M(EG)$

Pour tout (n, m) de $PP(EG)$

Si $F(EG, n, m)$ est vrai alors

$EG' = EG + \{(n, m)\}$

appel de match (EG')

Fin Si
 Fin pour tout
 Fin Si
fin Procédure match(EG)

où $F(EG, n, m)$ est une fonction booléenne qui retourne « vrai » lorsque le couple de sommets (n, m) de G et G' vérifient l'isomorphisme. Elle s'appuie sur l'utilisation des invariants, c'est à dire les conditions nécessaires mais pas suffisantes pour que deux sommets soient isomorphes.

2.4.4 Application à la chimie des macromolécules

Comme il a été dit précédemment il y a des analogies entre un graphe et une représentation moléculaire, le tableau 2.1 ci-après présente certains aspects de ces similitudes :

Tableau 2.1 : Similitudes entre un graphe et une représentation moléculaire

Graphe	Molécule
sommet	atome
graphe	structure
arbre	molécule acyclique
valence d'un sommet	valence d'un atome

Pour résoudre le problème de comparaison entre deux chaînes dans un système réactionnel, on peut utiliser les algorithmes d'isomorphisme entre deux graphes avec des modifications propres aux espèces chimiques susceptibles d'apparaître dans le système réactionnel à étudier. Ainsi dans notre cas, en plus d'implémenter l'algorithme mentionné précédemment, il a été choisi de prendre en considération les invariants suivants :

- Si deux chaînes sont identiques alors elles ont la même masse molaire;
- Si le sommet s_i est l'image par isomorphisme du sommet s_j alors ils ont la même composition chimique et le même nombre de successeurs et de précédents;
- Si l'arc (s_i, s_j) est identique à l'arc $(f(s_i), f(s_j))$ alors ils ont le même nombre d'unités répétitives (nombre de monomères).

Ces invariants sont très utiles pour réduire la complexité des calculs, car plus nous avons d'invariants moins le nombre de recours au programme de recherche d'isomorphisme entre deux graphes est requis, et conséquemment la complexité des calculs induits par le programme de simulation est réduite. Plusieurs autres invariants peuvent être déduits selon les cas étudiés.

2.5 Processus stochastiques markoviens

Lorsqu'on parle de modélisation stochastique, il est inévitable d'utiliser des notions de la probabilité et plus spécialement celles liées aux processus stochastiques. Dans notre cas nous allons nous intéresser aux chaînes de Markov.

2.5.1 Définition d'espace de probabilité

Un **espace de probabilité** est un triplet (Ω, F, P) où

- Ω est l'ensemble de toutes les réalisations ou **les trajectoires** associées avec une expérience. Ω est appelé l'**espace des trajectoires**.
- F est un ensemble de sous-ensembles de Ω tel que :
 1. $\emptyset \in F$ et $\Omega \in F$.
 2. si $A \in F$ alors l'ensemble complémentaire de A , noté A^c , est aussi dans F .
 3. si $A_n \in F$ pour $n = 1, 2, \dots$, alors $A_1 \cup A_2 \cup \dots \cup A_n \cup \dots$ appartient à F .

Un élément de F est appelé un **événement**. L'ensemble F est appelé une *tribu*.

- P est une **mesure de probabilité** sur (Ω, F) , c'est-à-dire que P est une application de F dans $[0, 1]$ telle que :

1. $P(\emptyset) = 0$ et $P(\Omega) = 1$.
2. $P(\cup_{n \in I} A_n) = \sum_{n \in I} P(A_n)$ pour n'importe quelle famille dénombrable (finie ou infinie) $\{A_n, n \in I\}$ d'événements 2 à 2 *disjoints* (i.e. $A_i \cap A_j = \emptyset$ pour $i \in I, j \in I$ tels que $i \neq j$).

2.5.2 Définition de processus stochastique

Un **processus stochastique** X_t est une suite de variables aléatoires indexées qui prennent des valeurs dans un ensemble $\mathcal{X}$, et où T est l'**ensemble des indices** t . Souvent t représente le temps. Si T est dénombrable on parle de **processus à temps discret**. Si l'ensemble des valeurs de T est continu, on parle de **processus à temps continu**. $\mathcal{X}$ est l'**ensemble des états du processus**. L'ensemble des états peut être continu ou discret.

En résumé, on peut dire que la caractéristique de base d'un processus stochastique est le fait que la loi de la variable X_t soit fonction de t . La notion de processus élargit la notion de variable aléatoire. Une réalisation d'un processus est appelée **trajectoire**. C'est donc la suite des réalisations des variables aléatoires X_t . Comme les réalisations d'une même variable aléatoire peuvent être différentes, les réalisations d'un même processus peuvent donner des trajectoires différentes.

Exemples :

1 : Le cours d'une action cotée en bourse au jour le jour et où X_t est la valeur de l'action à la date t ;

$T : \{\text{jour de cotation}\};$

$\mathcal{X} : \mathbb{R}^+.$

2 : Le nombre de molécules d'un réactif, X_t , dans un système réactionnel à la date t ;

$T : \{0 : \text{temps de réaction}\};$

$\mathcal{X} = \mathbb{N}$

2.5.3 Définition et caractéristiques des chaînes de Markov

- Une chaîne de Markov d'ordre k est un processus stochastique à temps discret et à espace d'états discrets caractérisé par le fait que tout le passé du processus est résumé dans les k états précédents.
- Une chaîne de Markov d'ordre 1 à J états est définie par les probabilités initiales d'être dans l'état j ($j = 1 \dots J$) et par la matrice des probabilités de transition entre les différents états.
- La caractéristique la plus importante d'une chaîne de Markov est constituée par l'évolution des probabilités de changement d'états en fonction du temps.
- Dans le cas où le temps est continu, on parle de processus de Markov.

2.5.4 Définition de processus de Markov

Un processus stochastique $(X_t, t \in [0, +\infty[)$ est dit processus de Markov continu avec espace d'états finis $E_X = \{1, \dots, K\}$ si et seulement si les transitions entre les états sont gouvernées par les intensités de transitions $\alpha_{hj}(t)$, ($h, j = 1, \dots, K$), tel que :

$$\alpha_{hj}(t) = \lim_{\Delta t \rightarrow 0} (\Pr \{X(t + \Delta t) = h \mid X(t) = j\} / \Delta t) \quad (2-6)$$

2.6 Modélisation des systèmes réactionnels chimiques

Les modèles utilisés pour représenter les systèmes réactionnels chimiques sont de différents types :

- Modèles probabilistes à événements discrets;
- Modèles à base de contraintes exprimées par des équations différentielles;
- Modèles à base de règles et lois statistiques.

Il est possible même parfois de recourir à l'utilisation de modèles hybrides, mélange de modèles précités. Généralement, un système réactionnel chimique est composé de n espèces chimiques ES_i , $i = 1, \dots, n$. On désigne par X_i le nombre de molécules de l'espèce ES_i et par V le volume du système. La valeur de $[ES_i] = X_i/V$ est appelée la concentration de l'espèce ES_i . Les modèles essaient de retracer l'évolution du système, ce qui revient à rapporter l'évolution du vecteur $([ES_1], \dots, [ES_n])$ en fonction du

temps ainsi que d'autres paramètres jugés utiles selon les cas étudiés, comme la distribution des masses lors de la polymérisation. Nous passerons en revue brièvement l'approche déterministe et l'approche statistique pour ensuite aborder plus en détail l'approche stochastique qui nous intéresse dans ce projet.

2.6.1 L'approche déterministe

L'approche déterministe propose des modèles faisant intervenir des équations différentielles pour décrire l'évolution des systèmes réactionnels chimiques. De par cette voie, elle suppose que les variables définissant le système sont des fonctions continues dans le temps. Dans le cas général le modèle a la forme suivante :

$$\begin{aligned} d[ES_1]/dt &= ff_1([ES_1], \dots, [ES_n]) \\ d[ES_2]/dt &= ff_2([ES_1], \dots, [ES_n]) \\ d[ES_n]/dt &= ff_n([ES_1], \dots, [ES_n]) \end{aligned} \quad (2-7)$$

Ce système si simple en apparence ne possède pas toujours de solutions analytiques. Souvent le recours aux méthodes numériques est indispensable pour pouvoir prédire l'évolution de la composition du système réactionnel. De plus, cette méthodologie ne peut prévoir aisément la structure des molécules du système réactionnel qui reste une information importante pour les réactions de polymérisation.

2.6.2 L'approche statistique

L'approche statistique propose de calculer les moments de la distribution des molécules formant le système réactionnel de polymérisation. Il s'agit souvent de calculer les premiers moments qui permettent de déduire certaines caractéristiques importantes quant à la prévision de l'état du système comme le degré de polymérisation et l'indice de polydispersité. Flory (1936) a été le premier à lier les valeurs de ces moments lors d'une réaction de polycondensation au degré d'avancement de la réaction pour le cas de chaînes linéaires. D'autres chercheurs dont Stockmayer (1943) et Macosko et Miller (1976), ont contribué au développement de cette approche. Avec les méthodes statistiques on peut prédire les moments de la distribution des poids molaires des

molécules sans toutefois arriver souvent à obtenir la structure exacte de ces molécules. Habituellement, on se limite aux premiers moments vu que d'une part, il est possible de les mesurer de façon expérimentale et que d'autre part les formules deviennent très complexes lorsque les molécules formant le système diffèrent par le nombre ou la nature des groupements fonctionnels. Nous constatons alors que cette approche n'est pas aussi efficace pour l'appliquer à toutes les réactions de polymérisation.

2.6.3 L'approche stochastique

L'approche stochastique permet d'obtenir plus de renseignements comparativement aux approches déterministe et statistique. Elle permet notamment d'avoir les informations sur la composition chimique du système et sur la structure des différentes molécules présentes dans celui-ci. Ce qui permet d'obtenir de meilleures estimations des caractéristiques du système.

Dans l'approche stochastique, l'évolution du système réactionnel est considérée comme le résultat d'une suite d'événements qui s'y produisent. Souvent ces événements sont les réactions chimiques qui s'y effectuent. A tout instant l'événement qui prend place dans le système est aléatoirement choisi parmi un ensemble d'événements susceptibles de se réaliser. Ainsi, l'état du système à n espèces chimiques dans le cas d'une évolution isotherme et à pression constante, est déterminé par les valeurs prises par le vecteur X défini par $X = (X_1, X_2, X_3, \dots, X_n)$ dont les composantes X_i , représentant le nombre des molécules de l'espèce i , sont des processus stochastiques. En effet, il est évident que les valeurs prises par le vecteur X entre deux instants t et $t + dt$ dépendent des réactions chimiques qui se sont produites dans le système dans l'intervalle dt . L'ordre et la fréquence de déroulement de ces réactions sont aléatoires. En effet, l'événement « la réaction R_j a eu lieu » est le résultat de plusieurs paramètres non nécessairement déterministes; Une réaction a lieu lorsque les molécules des réactifs se rencontrent et vérifient des conditions particulières comme avoir des énergies d'activation supérieures à un seuil déterminé. Comme nous venons de le voir, l'évolution du système présente des aspects aléatoires qui ne dépendent que de son état

présent. On retrouve par conséquent la définition des processus de Markov d'ordre 1. Le schéma présenté dans la figure 2.13 illustre cette vision de l'évolution du système réactionnel entre deux instants t et $t + dt$ avec dt suffisamment petit pour qu'un seul événement tienne place dans le système. Par abus de langage, lorsque la réaction R_j a eu lieu, on dit que le système a pris le « canal » R_j (Muruhan et al. (2003)).

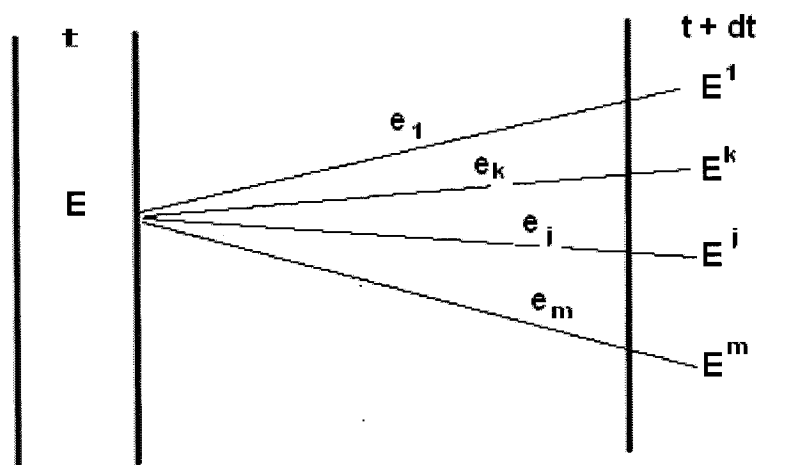
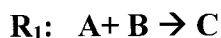


Figure 2.13 : Schéma représentant l'évolution du système réactionnel.

E est la valeur du vecteur d'état du système à l'instant t ;

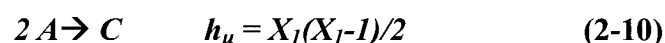
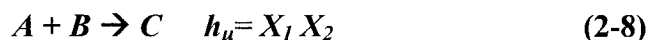
E^j est la valeur du vecteur état du système à l'instant $t + dt$, après que l'événement e_i ait eu lieu.

Chaque événement e_i a une probabilité qui lui est associée et qui dépend de la cinétique apparente des différentes réactions. Il est évident que parmi les facteurs influençant la probabilité d'occurrence d'une réaction, il y a le nombre de molécules des réactifs : plus grand est ce nombre, plus grande est la valeur de cette probabilité. Pour bien illustrer ce concept, considérons la réaction suivante :



Si c_1 désigne la probabilité par unité de temps que deux molécules des réactifs A et B vont réagir à l'instant t , et si nous avons a molécules de type A et b molécules de type B dans un volume V , alors la probabilité que la réaction R_1 se produise dans le volume V durant l'intervalle de temps $[t, t + dt]$ est égale au produit $a*b*c_1*dt$. En général, nous pouvons écrire que la probabilité pour qu'une réaction R_μ se produise dans

un système pendant l'intervalle de temps $[t, t + dt]$ est de la forme $c_\mu * h_\mu * dt$ (Gillespie et al. (1976)), avec h_μ le nombre de combinaisons de réactifs nécessaires pour que la réaction R_μ se produise. Ci-dessous les valeurs de h_μ pour les réactions simples les plus usuelles (Gillespie (1992)) :



Le produit $a_\mu = h_\mu * c_\mu$ est appelé la fonction intensité de probabilité et le produit $a_\mu * dt$ est appelé la probabilité que la réaction se produise dans le volume V entre les instants t et $t + dt$ étant donné que le système est dans l'état $(X_1, X_2, \dots, X_n)$ à l'instant t . Pour ce qui est des c_i , il y a une relation étroite entre ces coefficients stochastiques et les constantes de réactions chimiques déterministes k_i , comme le montre les relations pour les réactions simples suivantes :

- Pour une réaction de type : $S_1 \rightarrow S_2$, tel que

$$d[S_2]/dt = k[S_1],$$
nous avons $k = c$
- Pour une réaction de type : $S_1 + S_2 \rightarrow S_3$, tel que

$$d[S_3]/dt = k[S_1][S_2],$$
nous avons $k = c * N_a * V$ avec N_a le nombre d'Avogadro.
- Pour une réaction de type : $S_1 + S_2 \rightarrow S_3$, tel que

$$d[S_3]/dt = k[S_1]^2$$
nous avons $k = \frac{1}{2} c * N_a * V$

Dans un système où m réactions peuvent se produire nous aurons m constantes c_j , $j = 1, \dots, m$ qui ne dépendent que des propriétés physiques des molécules et de la température du système. Si l'on désigne par $p(x; t)$ la probabilité que le système

réactionnel chimique ait une population moléculaire $X = (X_1, \dots, X_n)$ à l'instant t , et si l'on désigne par v_u le vecteur de transition stoechiométrique lié à la réaction R_u , en se basant sur le raisonnement précédent, le vecteur X vérifie l'équation d'évolution dite « chemical master equation »:

$$p(x; t + dt) = p(x; t) \left(1 - \sum_{u=1}^m a_u(x) dt \right) + \sum_{u=1}^m p(x - v_u; t) a_u(x - v_u) dt \quad (2-11)$$

que l'on peut réécrire sous la forme différentielle suivante :

$$\frac{dp(x; t)}{dt} = -p(x; t) \sum_{u=1}^m a_u(x) + \sum_{u=1}^m p(x - v_u; t) a_u(x - v_u) \quad (2-12)$$

La résolution de cette équation différentielle n'est pas toujours facile, dans bien de cas le recours à la simulation stochastique est le seul moyen permettant d'avoir des estimations des solutions. En effet, si à l'instant t nous savons l'état du système réactionnel $X = (X_1, \dots, X_n)$, tout ce qu'on a besoin pour calculer l'état futur du système est de répondre aux questions suivantes :

Quand la prochaine réaction va se produire?

Quelle réaction va se produire?

Pour répondre à ces questions, les méthodes diffèrent, néanmoins nous pouvons les classer selon deux approches : La première suppose que les processus de réactions sont des processus de Poisson et conséquemment que les temps entre les réactions sont des variables aléatoires exponentielles, alors que la deuxième considère les états de repos du système, i.e. ceux pendant lesquels aucune réaction ne se produit, comme des événements particuliers à ajouter à l'ensemble d'événements susceptibles de se produire. Les algorithmes les plus importants et les plus cités sont:

2.6.4 Méthode de Gillespie

Gillespie est l'un des pionniers dans ce domaine et sa méthode s'inscrit dans la première classe. Voici une énumération des étapes principales de son algorithme (Gillespie (1977)) :

Étape 0 : (Initialisation)

- Lire les coefficients stochastiques $c_1, \dots, c_M$;
- Lire les nombres de particules $X_1, \dots, X_N$;
- Initialiser le temps de réaction t à 0;
- Initialiser le générateur de variables aléatoire unitaires

Étape 1 :

- Calculer $X_1, \dots, X_N$;
- Calculer les M probabilités des réactions $a_i = h_i * c_i$;

Étape 2 : Pour toute réaction R_μ

- Générer le temps d'attente τ_μ suivant la loi exponentielle de paramètre a_μ : $\tau_\mu = (1/a_\mu)\ln(1/URN)$;

Étape 3 :

- Mettre à jour la variable état du système après exécution de la réaction R_μ dont le temps τ_μ est le minimum des temps générés;
- Augmenter le temps de réaction t de τ_μ .

Étape 4 : (Terminaison)

- Stopper la simulation si t atteint le temps de terminaison T_f ;
- Retourner à l'étape 1 sinon.

2.6.5 Méthode de Gibson

La méthode de Gillespie est simple à implémenter, mais elle présente un temps d'exécution très grand, dû principalement aux mises à jour des variables du système

réactionnel. Plusieurs améliorations ont été proposées, parmi lesquelles on trouve la méthode de Gibson (Gibson et Bruck (2000)), dont l'idée est de minimiser le nombre des mises à jour des variables du système. Il s'agit de déterminer un graphe d'indépendance entre les composantes du système et les réactions pouvant y avoir lieu de façon à ne procéder qu'à la mise à jour des paramètres influencés par le déroulement d'une réaction R lorsque celle-ci a lieu dans le système réactionnel. Cette méthode s'avère utile lorsque le nombre de réactions à considérer est grand.

Étape 0 : (Initialisation)

- Lire les coefficients stochastiques $c_1, \dots, c_M$;
- Lire les nombres de particules $X_1, \dots, X_N$;
- Initialiser le temps de réaction t à 0;
- Initialiser le générateur de variables aléatoire unitaires URN;
- Générer le graphe de dépendance $G(S, A_S)$ où S est l'ensemble des sommets $S = (R_1, \dots, R_M)$ et A_S l'ensemble des arcs liant les sommets. Les arcs orientés sont établis pour toute paire de réactions (R_μ, R_α) tel que l'exécution de R_μ change la fonction de probabilité de R_α . Un exemple de ce graphe de dépendance est illustré par la figure 2.14.
- Pour chaque réaction R_μ , générer le temps d'attente τ_μ suivant la loi exponentielle de paramètre a_μ

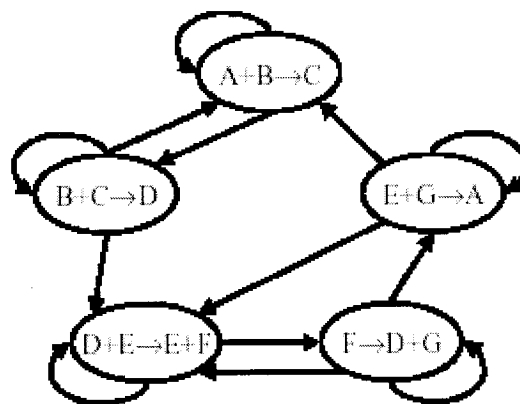


Figure 2.14: Exemple de graphe de dépendance de la méthode de Gibson

Étape 1:

- Déterminer le μ tel que τ_μ est le minimum des temps d'attentes générés;
- Augmenter le temps de réaction t de τ_μ .

Étape 2:

- Mettre à jour les variables d'état après exécution de R_μ .

Étape 3:

- Mettre à jour les a_α pour toute paire (R_μ, R_α) dans le graphe de dépendance G .
 si $\alpha \neq \mu$ alors $\tau_\alpha = (a_{\alpha, \text{ancien}}/a_{\alpha, \text{nouveau}}) (\tau_\alpha - t) + t$
 si $\alpha = \mu$ alors $\tau_\alpha = r + t$, r suit l'exponentielle de paramètre a_μ ;

Étape 4: (Terminaison)

- Arrêter la simulation si t atteint le temps de terminaison Tf ;
- Retourner à l'étape 1 sinon.

2.6.6 Méthode de Pandit

Dans la deuxième classe, on trouve la méthode de Pandit (1992) qui a étudié les réactions de polycondensation en considérant les temps de repos du système comme un événement à ajouter à l'ensemble des événements susceptible de prendre place dans le système. Il lui a assigné une probabilité d'occurrence égale à p^* , inversement proportionnelle à une valeur de temps t^* qui désigne le temps que le système passe au repos. La valeur de t^* est choisi arbitrairement petit. En tenant compte de ces hypothèses, il a proposé l'ossature de l'algorithme suivant pour suivre l'évolution du système réactionnel:

Étape 0:

- Lire les coefficients stochastiques $c_I, \dots, c_M$, et de t^* ;
- Lire les nombres de particules $X_I, \dots, X_N$;
- Initialiser le temps de réaction t à 0;
- Initialiser le générateur de variables aléatoire unitaires URN.

Étape 1:

- Calculer $X_1, \dots, X_N$;
- Calculer les M probabilités des réactions $a_i = h_i * c_i$;
- Générer une variable aléatoire unitaire u ;
- Ordonner les valeurs $1/t^*, a_i, i = 1, \dots, M$ dans un vecteur PV ;
- Choisir l'événement k correspondant au premier indice du vecteur PV tel que $u < (\sum_j^k PV(j)) / (\sum_j^M PV(j))$.

Étape 2:

- Augmenter le temps de réaction t de t^* si l'événement choisi est l'événement « système au repos » et aller à l'étape 3;
- Mettre à jour les variables d'état du système selon la réaction R_u choisie.

Étape 3: (Terminaison)

- Arrêter la simulation si t atteint le temps de terminaison Tf ;
- Retourner à l'étape 2 sinon.

Pour remédier au problème de complexité des calculs et au temps d'exécution de sa méthode, Pandit propose de prendre un facteur d'échelle fe pour les réactions, ce qui revient à remplacer l'événement « R_u a lieu » par l'événement « $1/fe$ réactions R_u ont eu lieu ». Ainsi plus est grand fe , plus est petite la quantité de molécules qui réagit à chaque événement. Dans ce travail nous avons implémenté la méthode de Pandit qui a traité la simulation stochastique des polymérisations des systèmes linéaires, un sujet qui est proche de celui de notre étude.

Comme nous venons de le voir, plusieurs méthodes ont été proposées pour réaliser la simulation stochastique d'un système réactionnel chimique. La méthode de Gillespie est simple à implémenter, elle est recommandée pour des systèmes théâtre d'un nombre de réactions petit et dont l'interdépendance entre les réactions est forte. Celle de

Gibson apporte des améliorations visant à réduire le nombre de calculs et par la suite le temps d'exécution de la méthode de Gillespie. Elle est recommandée lorsque le nombre de réactions est grand et l'interdépendance entre les réactions est faible. La méthode de Pandit considère l'avancement du temps comme un événement en soi et introduit un facteur d'échelle pour réduire le nombre de calcul et traiter le problème des événements rares. Le point commun entre elles est qu'elles considèrent l'évolution du système comme une suite d'événements qui s'y produisent, le nombre de ces événements peut différer selon le modèle pris pour représenter le système à étudier. En effet, on ne peut considérer toutes les réactions qui se produisent lors de l'évolution d'un système complexe, la tendance est de déterminer les réactions qui jouent un rôle déterminant dans son évolution. Dans notre cas, qui est la polymérisation par ouverture de cycle, nous allons proposer notre modèle et proposer une méthodologie pour déterminer les coefficients stochastiques régissant les événements constituant ce modèle. Nous allons proposer une représentation des chaînes qui permettra de faciliter l'étude des polymérisations moyennant l'approche stochastique. Cette représentation permettra de résoudre le problème de reconnaissance des chaînes rencontré souvent lors de la simulation stochastique des réactions de polymérisation. De plus dans ce travail, nous allons proposer une méthode pour l'estimation des constantes de vitesse des réactions principales se produisant lors de la polymérisation par ouverture de cycle. Cette méthode repose sur la réduction des disparités entre les données expérimentales et celles issues du modèle proposé. Cette étude élargit ainsi les utilisations de l'approche stochastique pour permettre non seulement la prévision de l'évolution d'une réaction de synthèse par ouverture de cycle, mais aussi l'estimation des constantes de vitesse de réactions. Ainsi en partant de l'état microscopique d'un système réactionnel chimique nous avons pu établir les liens nous permettant d'en déduire l'évolution dans le temps de ses paramètres macroscopiques. Par cela, cette étude a permis d'élargir les champs d'application de l'approche stochastique dans l'étude des polymérisations par ouverture de cycle. Le chapitre suivant exposera plus en détail les démarches et les étapes entreprises pour réaliser ces objectifs.

CHAPITRE III: MÉTHODOLOGIE

Dans ce chapitre nous allons explorer la méthodologie que nous avons adoptée pour réaliser ce projet. Il sera question notamment de faire le point sur le formalisme de modélisation de la polymérisation des lactides en présence d'octoate d'étain et d'alcool à différentes fonctionnalités $R(OH)_x$, de montrer la représentation du système réactionnel et celles des chaînes de polymère, d'exposer l'idée de base ayant servi à estimer les constantes de vitesse des réactions principales, de décrire l'exécution du programme et enfin d'énumérer les logiciels et le matériel informatique utilisés. Ainsi le chapitre est organisé selon les sous-sections suivantes :

- objectif;
- modélisation de la polymérisation des lactides;
- estimation des constantes de vitesse des réactions;
- représentation du système réactionnel;
- description du déroulement de l'application;
- moyens logiciels et matériels.

3.1 Objectif

L'objectif de ce projet est de réaliser une application qui par le biais de la simulation stochastique, pourra prévoir l'évolution et estimer les constantes de vitesse des principales réactions se produisant dans un système réactionnel. Comme nous l'avons annoncé dans le premier chapitre, le choix de la réaction de polymérisation des lactides n'est pas restrictif. Il est fait pour élucider les idées. L'application pourra, moyennant certaines modifications, traiter d'autres réactions de polymérisation. Ce souci est un des facteurs qui nous a motivés à conserver le caractère général lors de la réalisation de ce projet et de proposer par la suite des solutions qui dépassent le cadre de la polymérisation des lactides. Ainsi, à titre d'exemple, la représentation des espèces chimiques utilise le modèle des graphes au lieu d'utiliser des structures plus simples qui

resteraient limitées aux cas étudiés sans possibilité d'utilisation ultérieure dans un système réactionnel différent. Cette vision a été respectée autant que possible dans le but de faciliter les développements futurs des prochains travaux.

3.2 Modélisation de la polymérisation des lactides

La réaction que nous allons étudier est la polymérisation des lactides en présence d'octoate d'étain et d'alcools à différentes fonctionnalités $R(OH)_x$. Comme dans la plupart des réactions de polymérisation, nous avons considéré le système réactionnel chimique comme étant le théâtre de trois réactions principales: initiation, propagation et terminaison. De plus, nous avons adopté le mécanisme déduit de ceux présentés dans les articles de Kricheldorf et al. (1995) et Nijenhuis et al. (1995), ce qui revient à considérer les réactions suivantes :

- **Initiation** : l'initiation peut être représentée par la réaction schématisée dans la figure 3.1 où R' n'est autre que $R-(OH)_{x-1}$. Ainsi, un seul site réactionnel correspondant à une liaison $R-OH$ est initialisé. La constante de vitesse de réaction est k_i .

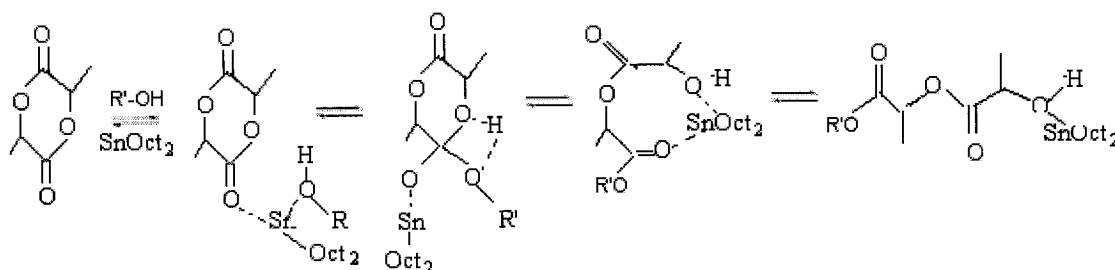


Figure 3.1 : Initiation de la polymérisation des lactides.

- **Propagation** : la réaction de propagation obéit à l'équation de récurrence formalisée dans la figure 3.2 pour le cas particulier où $x=3$. Il y a alors trois sites réactionnels possibles. Les degrés de polymérisation de chaque branche avant la propagation sont n , m et d , qui deviendront après la réaction n' , m' et d' de façon à vérifier l'égalité $n'+m'+d' = n+m+d+1$ qui

traduit l'effet de l'ajout de deux molécules d'acide lactique suite à l'ouverture de cycle de la molécule de lactide. La constante de réaction k_p est fonction du degré de polymérisation total soit $n + m + d$.

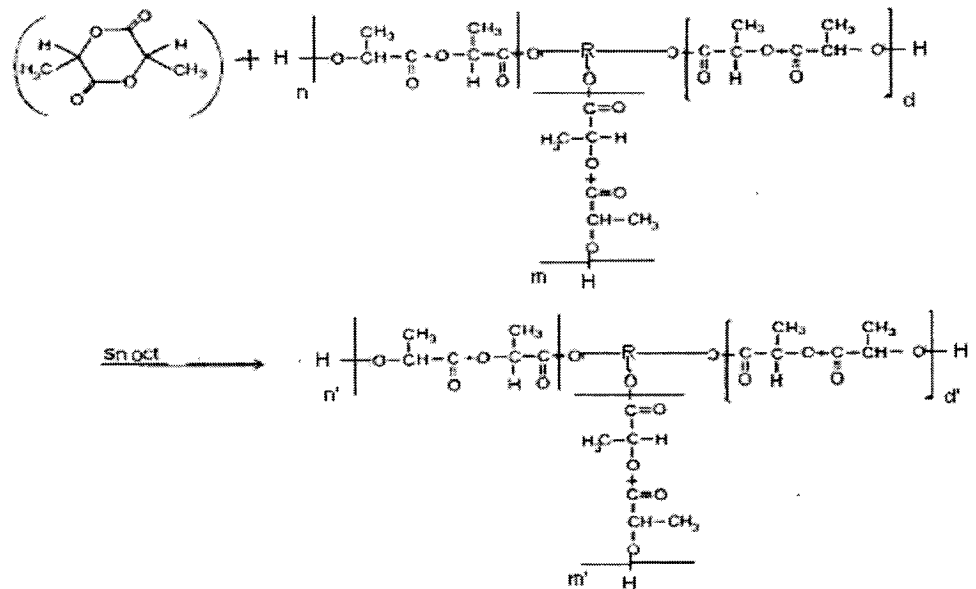


Figure 3.2 : Schéma de propagation d'une chaîne de polylactide co-initiée par un triolcool.

- **Terminaison** : la réaction de terminaison se traduit par la fermeture d'un site réactionnel, une liaison OH ne sera plus disponible pour une propagation éventuelle. La constante de vitesse de la réaction est k_t .

L'approche stochastique basée sur la représentation markovienne non homogène d'un système réactionnel repose sur la définition de l'état de ce système, de l'ensemble des événements susceptibles de s'y produire et des équations définissant l'évolution des probabilités de transition. Pour cela nous avons adopté les notations suivantes :

$X_1(t)$: nombre de monomères à l'instant t

$X_2(t)$: nombre de sites réactionnels actifs à l'instant t

$X_3(t)$: nombre de sites initialisés jusqu'à l'instant t

tab : tableau des chaînes présentes dans le système à l'instant t .

Dans le cas d'une évolution isotherme et isobare, le système est parfaitement défini par la connaissance de ces quatre structures. Si on suppose de plus que les sites réactionnels sont equiréactifs et que les réactions de propagation ont des constantes de vitesse de réactions k_p identiques donc indépendantes du degré de polymérisation, la liste des événements susceptibles de se produire dans le système selon la méthode de Pandit (Pandit et al. 1992) sera formée des événements suivants :

- Événement 1 : Initiation
- Événement 2 : Propagation
- Événement 3 : Terminaison
- Événement 4 : Système au repos.

À chacun de ces événements nous avons attribué une probabilité d'occurrence qui est fonction de l'état du système et des coefficients stochastiques. Pour cela nous avons adopté le modèle cinétique suivant :

$$d[{-OH^*}]/dt = k_i [{-OH}] [M] \quad \text{initiation} \quad (3-1)$$

$$d[{-M{-OH}}]/dt = k_p [M] [{-OH^*}] \quad \text{propagation} \quad (3-2)$$

$$d[{-OH_T}]/dt = k_t [{-OH^*}] \quad \text{terminaison} \quad (3-3)$$

où

$-OH^*$ désigne un site réactionnel actif

$-OH$ désigne un site réactionnel non initialisé

$-OH_T$ désigne un site réactionnel qui a subi une terminaison

$-M{-OH}$ désigne une chaîne

M désigne le monomère

En exprimant ces équations en fonction des nombres de particules et après simplification nous avons :

$$c_i = k_i / V; \quad (3-4)$$

$$c_p = k_p / V; \quad (3-5)$$

$$c_t = k_t; \quad (3-6)$$

où V est le volume du système réactionnel.

Et en termes de fonctions de probabilité :

$$p_1 = c_i * X_1(t) * (N_0 - X_3(t)); \quad (3-7)$$

$$p_2 = c_p * X_1(t) * X_2(t); \quad (3-8)$$

$$p_3 = c_t * X_2(t); \quad (3-9)$$

$$p_4 = 1/t^*; \quad (3-10)$$

où N_0 est le nombre initial de sites non initialisés qui est fonction du nombre initial des co-initiateurs.

À ce stade, le modèle représentant la réaction de synthèse étudiée est parfaitement défini. Nous allons discuter maintenant des altérations intervenant sur la composition chimique du système et sur le temps de réaction suite à l'occurrence de ces événements :

- **Initiation** : il y a ajout d'un site réactionnel, ce site pourra être un site non activé d'une molécule multi-chaînes déjà existante dans le système ou un site réactionnel d'une nouvelle molécule qui sera introduite dans le système.
- **Propagation** : il y a ajout d'un monomère sur une chaîne existante dans le système. Tous les sites réactionnels étant equiréactifs, le choix de la chaîne découlera du choix du site. Ce dernier sera choisi aléatoirement parmi tous les sites réactionnels actifs présents dans le système.
- **Terminaison** : un site réactionnel parmi les sites réactionnels actifs présents dans le système sera désactivé, son choix est aléatoirement choisi.
- **Système au repos** : le temps de réaction sera incrémenté de t^* , valeur choisie pour caractériser le temps que le système passe au repos.

À l'aide de ces définitions, nous avons donc proposé les paramètres servant à suivre l'évolution du système à partir d'un instant t où son état est connu. En effet, pour prévoir l'évolution du système, il suffit de réaliser la simulation Monte-Carlo selon la méthode de Pandit. En outre, vu la relation étroite entre les coefficients stochastiques du modèle et les constantes de vitesse des réactions, nous pouvons même réaliser des estimations de ces constantes en s'appuyant sur les informations obtenues de données expérimentales de la réaction de synthèse. Ce point sera discuté dans la section suivante.

3.3 Estimation des constantes de réactions

Le déroulement de la simulation de l'évolution d'un système réactionnel permet d'estimer les coefficients stochastiques à partir de cinétiques expérimentales connues. Ainsi, les constantes de réaction s'obtiennent en minimisant une fonction objective f , fonction des résultats de la simulation et des sources d'information dont nous disposons. Le problème ressemble à un problème classique d'apprentissage, qui est celui de trouver la fonction optimale dans un ensemble de fonctions paramétrées représentant le système d'intérêt. Dans le cas d'une source d'information $I(t)$ définie sur N observations, et d'un processus stochastique $Y(t)$, disons à titre d'exemple la conversion, nous avons à chercher $c=(c_b, c_p, c_t)$ de façon à minimiser $f_c(Y, I)$ définie par :

$$f_c(Y, I) = \sum_i^N (E(Y_i) - I_i)^2 \quad (3-11)$$

où $E(Y_i)$ désigne l'espérance de $Y(i)$

I_i désigne la i ème observation dans le temps de la source d'information $I(t)$

Cette formulation pourra être généralisée dans le cas de L sources d'information représentées par un vecteur I , $Y(t)$ sera un vecteur de variables aléatoires représentant les mêmes informations que celles contenues dans I . Ainsi, nous aurons :

$$f_c(Y, I) = \sum_l \alpha_l \sum_i^N (E(Y_{li}) - I_{li})^2 \quad \text{avec } \sum_l \alpha_l = 1 \quad (3-12)$$

L'approche retenue est donc d'utiliser les informations que nous avons pour déduire les coefficients stochastiques. Ces données expérimentales peuvent comprendre non seulement la conversion chimique du monomère, mais aussi des sources d'informations sur la structure des chaînes comme les moments de la distribution des masses molaires. Les poids α_i assignés aux différentes sources permettent de relativiser l'importance d'une source vis à vis des autres. De plus, des facteurs multiplicatifs peuvent être requis pour normaliser et rendre les différentes variables comparables entre elles. À titre d'exemple, les données de conversion sont comprises entre 0 et 1 alors que celles de M_n sont largement supérieures à 1. Ainsi la division de ces dernières par une constante les rendant entre 0 et 1 s'avère nécessaire si l'on veut utiliser ces deux sources dans une même fonction objective. Une fois que les coefficients stochastiques sont estimés, les constantes de vitesse des réactions le sont aussi tel qu'exprimé par les équations (3-4), (3-5) et (3-6). Dans notre cas nous avons utilisé une fonction définie sur deux sources d'information, soient la conversion et M_w . Notons que plusieurs fonctions objectives peuvent être utilisées, comme chercher à minimiser la différence entre les deux courbes $X(t)$ et $Y(t)$ en termes de surfaces, ce qui revient à considérer f tel que : $f = \int |E(Y) - I|$ qui est similaire aux définitions présentées précédemment si le pas d'intégration est constant.

3.4 Représentation du système réactionnel

Dans ce projet, la réaction de synthèse des polylactides est simulée par une approche stochastique. Un des avantages de cette approche est qu'on peut avoir des renseignements à l'échelle microscopique, i.e. à l'échelle de molécules. Cependant, afin d'obtenir les données sur la structure et de suivre l'évolution du système réactionnel comme étant une suite d'événements faisant intervenir les différentes espèces chimiques, il faut conserver les renseignements sur toutes les molécules présentes dans le système à un instant t donné. Par conséquent, un codage adéquat de l'information s'avère essentiel étant donné le nombre de molécules existant dans un système réactionnel. Pour ce faire, nous avons opté pour des hypothèses simplificatrices tout en préservant l'analogie entre

un graphe et une molécule, déjà décrite dans le deuxième chapitre. Pour illustrer notre propos, prenons l'exemple de chaînes produites selon la réaction représentée par le schéma de la figure 3.3, une application immédiate du principe de la théorie des graphes nous donnera un graphe avec un nombre de sommets supérieur à $3p$ où p est le degré de polymérisation de chaque branche, ce qui causera beaucoup de calculs lors de la comparaison de deux chaînes de ce type vu que p est souvent très grand. Pour minimiser le nombre de calculs il faut minimiser le nombre de sommets, pour cela il suffit de remarquer que l'on peut représenter la même chaîne avec un nombre minimal de sommets en changeant les caractéristiques de la similitude graphe-molécule définie dans le deuxième chapitre. En effet, plutôt que de considérer la règle stipulant que chaque sommet correspond à un atome, nous allons adopter celle stipulant que chaque sommet correspond à une molécule $C_xH_yO_z$ faisant partie de la chaîne. On retrouve là l'idée de base de la notion de pseudographe introduite par Gordon et Temple (1975). Cette idée vise à représenter une molécule comme un noyau central lié à des noyaux représentant ses groupements fonctionnels. Nous avons développé cette idée pour l'utiliser dans notre projet.

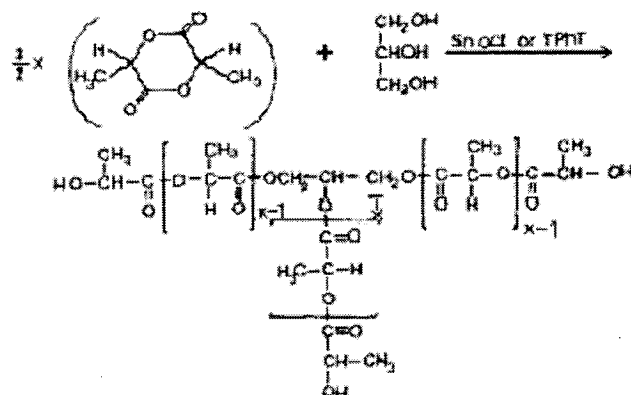


Figure 3.3 : Molécule de polylactide initiée par un glycolide.

En se basant sur cette règle, le codage repose sur le fait que la chaîne est un ensemble de noyaux liés entre eux par des liaisons actives et fixes. Lorsque la liaison peut réagir on parle de liaison active, comme par exemple la liaison double entre les

noyaux $\text{CH}_2=\text{CH}_2$ tandis qu'on parle de liaison fixe dans le cas contraire, comme par exemple CH_3-CH_3 . Cette définition apporte l'hypothèse simplificatrice où les réactions internes des chaînes ne sont pas considérées. La chaîne est donc traitée comme un ensemble de noyaux liés entre eux tel que les liaisons actives (terminales) subissent des ajouts de monomères comme le montre la figure 3.4 :

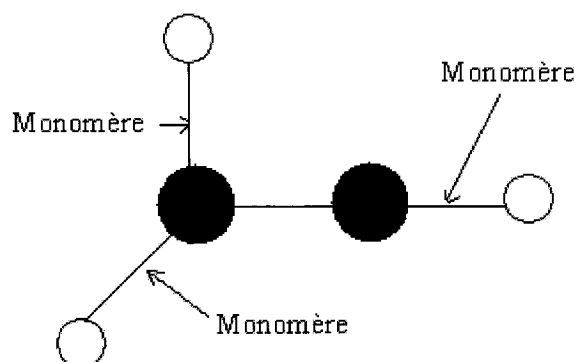


Figure 3.4 : Modèle d'une chaîne.

Une autre caractéristique qui va pouvoir nous permettre d'utiliser à fond le modèle simplifié est le fait que le monomère issu de l'ouverture de cycle de la molécule de lactide est symétrique. Ainsi il suffit de savoir le nombre de monomères ajoutés sur les liaisons actives en plus du type de la chaîne initiale pour déduire la structure de la chaîne. Pour les autres cas où le monomère issu n'est pas symétrique, par exemple pour du ϵ -caprolactone, nous pouvons adopter le même codage si l'intérêt se rapporte uniquement aux calculs relatifs à la distribution des masses molaires ou bien de définir un codage adéquat pour une liaison active pour tenir compte des différents types de monomères qui peuvent s'y ajouter. Ainsi dans le cas de deux types de monomères nous pouvons utiliser une chaîne de caractères uniquement formée des deux lettres *D* et *F*. La séquence de ces dernières dans la chaîne de caractères, représentant la liaison active, nous renseignera sur la séquence des deux types de monomères sur celle-ci.

Arrivé à ce point, nous pouvons dire qu'une chaîne est complètement définie si l'on connaît le nombre de monomères sur les liaisons actives et le type d'initiateur, ou la chaîne qui est à l'origine de sa formation. Cette dernière information nous permet de

reconstruire la chaîne puisqu'elle nous aide à constituer les noyaux et les liaisons existants entre les différents noyaux. Ainsi une chaîne est codée par le type de l'initiateur, ou une molécule dans d'autres cas, à partir duquel elle est formée et le tableau d'entiers représentant les nombres de monomères qu'elle contient sur ses liaisons actives.

La nouveauté que nous apportons ici est de pouvoir subdiviser la chaîne en blocs de molécules de façon à réduire le nombre total des sommets tout en préservant les caractéristiques requises pour les utilisations subséquentes de l'information. Notons que cette décomposition de la chaîne en blocs et liaisons est très importante et délicate. Il faut la faire de façon éclairée car une décomposition maladroite pourrait amener à considérer deux chaînes comme identiques alors qu'en réalité elles ne le sont pas. L'état des liaisons entre les blocs est rapportée par une matrice ML définie par :

$$ML = (a_{ij}) \text{ où } \begin{aligned} a_{ij} &= -2 \text{ si la liaison est fixe} \\ &= 1 \text{ si la liaison active est initialisé} \\ &= 0 \text{ si la liaison active est non initialisé} \\ &= -1 \text{ si la liaison active a subi une terminaison} \end{aligned} \quad (3-13)$$

Avec cette dernière définition le problème de comparaison de deux chaînes est résolu. En effet, deux chaînes sont identiques si elles ont la même structure et si les états des liaisons correspondantes sont égaux.

3.5 Déroulement d'une simulation

Cette section va exposer l'utilisation du code de simulation. C'est grâce à ce code que l'on peut prévoir les besoins et les résultats escomptés du développement de notre projet. Ainsi, pour pouvoir utiliser l'application, il faut commencer par entrer les données nécessaires pour accomplir la tâche qui lui est confiée. Ces données dépendent du but choisi par l'utilisateur, soit la prévision de l'évolution du système réactionnel chimique ou bien l'estimation des constantes de vitesses réactionnelles. Les données sont

contenues dans un fichier de nom 'entrees.txt'. Cette solution permet aisément de modifier la nature des calculs effectués, de ne pas oublier certains paramètres et enfin de mémoriser les données en vue d'une exploitation ultérieure. Plus spécifiquement, ce fichier, (un exemple est sur l'annexe 1), contient les paramètres définissant le système chimique réactionnel et les noms des fichiers résultants:

- Action à entreprendre (prévision ou estimation);
- Les coefficients stochastiques des réactions;
- Les pas de déplacement pour l'optimisation;
- Le nombre de monomères initiaux;
- Le nombre d'initiateurs initiaux;
- Le type alcool utilisé;
- Le critère d'arrêt pour les simulations (nombre de simulations ou précision sur les moments).

Une fois que le programme a lu ces paramètres, il commence selon l'action choisie à exécuter les séquences nécessaires pour atteindre son objectif. Dans le cas où l'action désirée est la prévision de l'évolution du système réactionnel chimique, le programme va faire un nombre n de simulations de façon à obtenir des prévisions statistiquement significatives, tandis que dans le cas où l'action entreprise est d'estimer les constantes des vitesses des réactions, le programme va chercher les valeurs optimales pour le problème décrit précédemment. Il s'agit de chercher la solution (c_1, c_2, c_3) qui minimisera les différences entre les estimations issues des simulations et les résultats expérimentaux. Comme on peut le remarquer, dans les deux cas le processus prédominant est celui de la simulation stochastique et de ce fait le noyau de l'application est le déroulement de la simulation. Pour ce faire, l'application effectue une suite d'événements sélectionnés selon les probabilités qui leur sont assignées. Toutefois, pour des considérations de temps de calcul et d'utilisation optimale de la mémoire, chaque fois qu'il y a transformation d'une chaîne suite à une initiation, une propagation ou une terminaison, cette dernière est placée de façon à préserver l'ordonnancement des chaînes par ordre croissant de leur poids moléculaire. Ce triage est important puisqu'il permet de

mémoriser les structures des chaînes identiques une seule fois. De plus, la comparaison des chaînes lors des choix d'événements se fait ainsi au sein d'un tableau moins grand.

Certains résultats de l'application sont présentés sous forme de graphiques traçant l'évolution des paramètres jugés importants et de fichiers de données qu'on peut exploiter ultérieurement. La solution d'enregistrer les résultats dans des fichiers s'avère intéressante notamment lors de l'estimation des constantes de vitesse des réactions. En effet, comme il s'agit d'une recherche de point minimale la procédure n'est pas toujours convergente. Aussi il faut repartir de nouveau avec des constantes de départ tout en s'inspirant des résultats obtenus. L'application génère plusieurs fichiers de résultats, énumérés ci-après, avec une description détaillée de leur contenu:

- Fichier 'optimisation.txt' : contient les données sur les points (c_b, c_p, c_t) visités lors de l'estimation des constantes de vitesse des réactions avec les valeurs prises par la fonction objective en ces points. Un extrait du contenu de ce fichier est à l'annexe 2.
- Fichier 'polyfinal.txt' : contient les données sur le système initial et pour chacune des n simulations faites, un résumé de la composition finale du système. Un extrait du contenu de ce fichier est à l'annexe 3.
- Fichier 'proba.txt' : contient les données sur le système initial et pour chacune des n simulations les données relatives à l'évolution des probabilités d'occurrence des événements susceptibles de se produire dans le système. Un extrait du contenu de ce fichier est à l'annexe 4.
- Fichier 'moments.txt' : contient les données sur le système initial et pour chacune des n simulations les données relatives à l'évolution des moments de la distribution de la masse molaire M_n, M_w, M_z . Un extrait du contenu de ce fichier est à l'annexe 5.

En plus de ces fichiers l'application génère des graphiques dans le but de présenter des résultats préliminaires sur le déroulement des simulations et en montrant l'évolution du système:

- Graphique de conversion : présente l'évolution de la conversion des monomères en fonction du temps et dans le cas où l'utilisateur dispose de résultats expérimentaux, une superposition des deux courbes de conversion est affichée.
- Graphique de probabilités : retrace l'évolution dans le temps des probabilités des événements susceptibles de se produire dans le système.
- Graphique de distribution de masse : trace la distribution massique molaire des chaînes contenues dans le système à l'instant d'arrêt.
- Graphique des moments : trace l'évolution des trois premiers moments de la distribution massique M_n , M_w , M_z en fonction du nombre de simulations effectuées.
- Graphique de M_w : retrace l'évolution dans le temps du deuxième moment de la distribution massique du système et dans le cas où l'utilisateur dispose de résultats expérimentaux, une superposition des deux courbes de M_w simulée et expérimentales est affichée.

Les solutions retenues pour résoudre un problème dépendent des moyens disponibles et le génie du logiciel n'échappe pas à cette règle. Le programme élaboré a été développé en utilisant le logiciel de développement Pro Fortran 8.2 pour Windows d'Absoft Corporation et il a été tourné et testé en utilisant un ordinateur portable de type Toshiba. Les ordinateurs portables récents possèdent de puissantes performances en termes de mémoire et de calcul. Toutefois, nous signalons que pour avoir de plus en plus de précision sur les résultats et pour gagner du temps, l'utilisation de grandes machines n'est pas à écarter. D'autre part, notre choix pour le Fortran comme langage de développement est justifié par le fait que c'est un langage scientifique de choix vu l'existence de modules prêts à être incorporés notamment pour le problème

d'optimisation que nous avons abordé lors de l'estimation des constantes de vitesse des réactions.

Dans ce chapitre, il a été question de la méthodologie que nous avons adoptée pour réaliser ce projet. Dans le chapitre suivant, nous allons voir comment nous avons procédé pour élaborer les modules de cette application ainsi que les résultats que nous avons obtenus en l'utilisant pour étudier la polymérisation des lactides en présence d'octoate d'étain et d'alcool à différents nombre de fonctionnalité $R(OH)_x$.

CHAPITRE IV: ÉLABORATION DE L'APPLICATION ET RÉSULTATS DE L'ESTIMATION DE LA SYNTHÈSE DES POLYLACTIDES.

Dans ce chapitre il est question d'aborder les principales étapes de l'élaboration de notre application dont le double objectif est d'estimer les constantes de vitesse des principales réactions chimiques se produisant dans une polymérisation par ouverture de cycle et/ou de simuler l'évolution de ce système, connaissant les constantes cinétiques. Le cas particulier traité dans ce mémoire pour valider et illustrer ce développement est la réaction de synthèse des polylactides en présence de l'octoate d'étain et d'alcools à différents nombre de fonctionnalités $(R-OH)_x$.

Comme il a été mentionné dans le chapitre précédent, nous avons fait le choix de développer l'application en langage Fortran 90 à l'aide du compilateur ProFortran 8.2 d'ABSOFIT. Ce langage présente des aspects de programmation fonctionnelle et d'autres de programmation orientée objet avec une nette tendance pour la première approche. Pour cette raison, nous avons choisi d'adopter l'approche structurée dans l'analyse servant au développement de l'application tout en essayant de garder certains aspects de l'approche orientée objet lorsque c'est possible. Ce chapitre est constitué des sections suivantes: analyse fonctionnelle des besoins, algorithmes et structures de l'information, conception de l'application et application à la synthèse des polylactides.

4.1 Analyse fonctionnelle des besoins

Le but de cette étape est la spécification des besoins et requis attendus de l'utilisateur final de l'application à l'aide de schémas et de modèles en se servant de l'analyse fonctionnelle. Il a été décidé de modéliser ces attentes moyennant des graphes, facilitant la tâche de l'analyste. Dans cette partie, nous allons procéder à l'élaboration du modèle entité relation ERD et des diagrammes de décomposition des flux des données DFD.

4.1.1 Modèle entité association

Ce modèle a pour but d'énumérer les entités qui font partie de l'application ainsi que les associations qui existent entre elles.

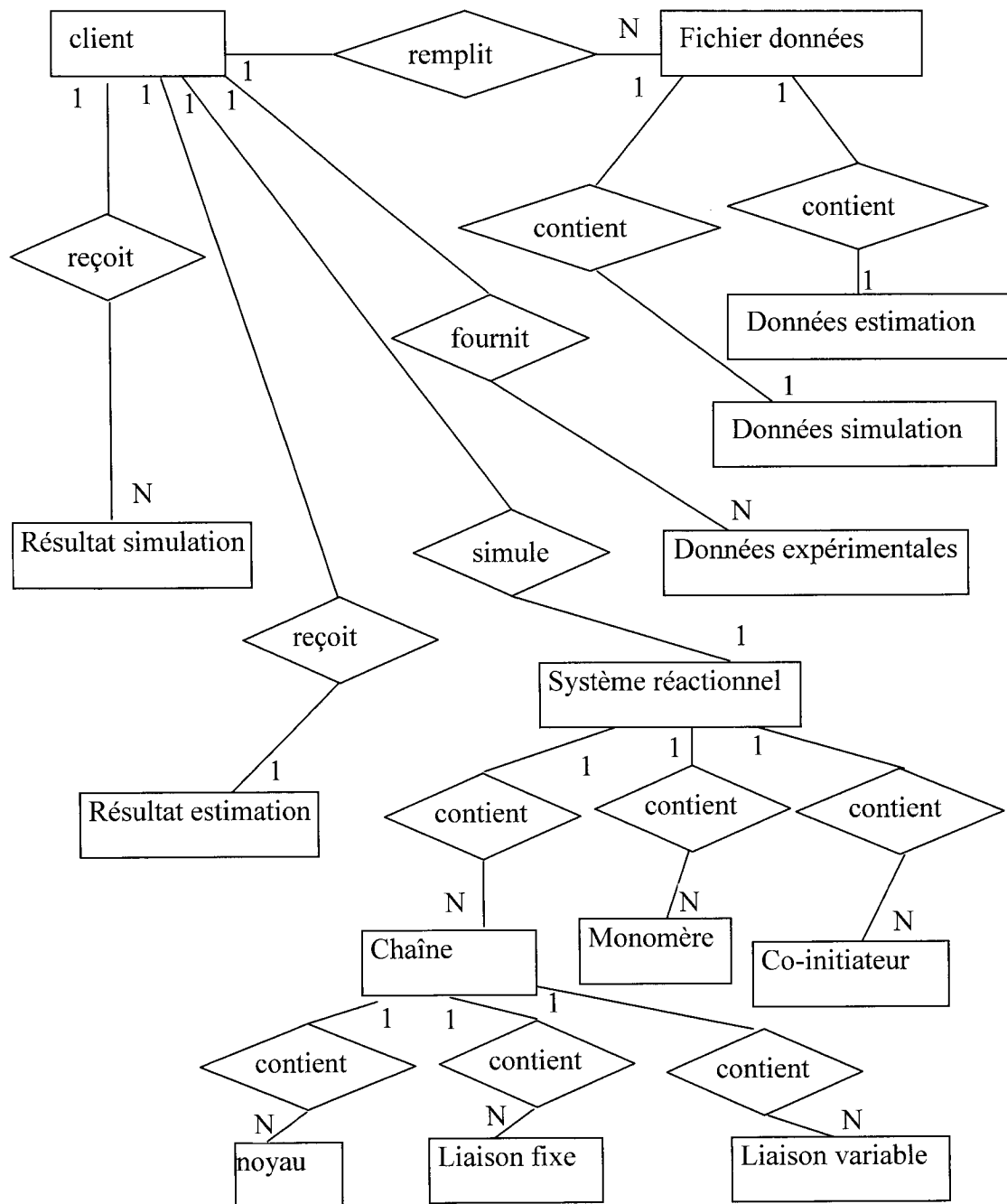


Figure 4.1 : Diagramme entité association.

4.1.3 Les diagrammes de flux des données :

L'idée de base des diagrammes de flux des données est de voir le déroulement de l'application sous l'angle d'échanges de flots des données entre les différents processus qui la composent. Le principe est de décomposer les processus agissant sur les flux de données itérativement jusqu'à aboutir à des processus élémentaires ou jugés ainsi par l'analyste. Dans notre cas, nous allons commencer par le diagramme de contexte qui vise à définir le but de l'application à développer et les résultats espérés par son développement. Ensuite il y aura des décompositions de niveaux inférieurs dont le rôle est d'identifier les sous-processus adjacents au processus de niveau supérieur immédiat.

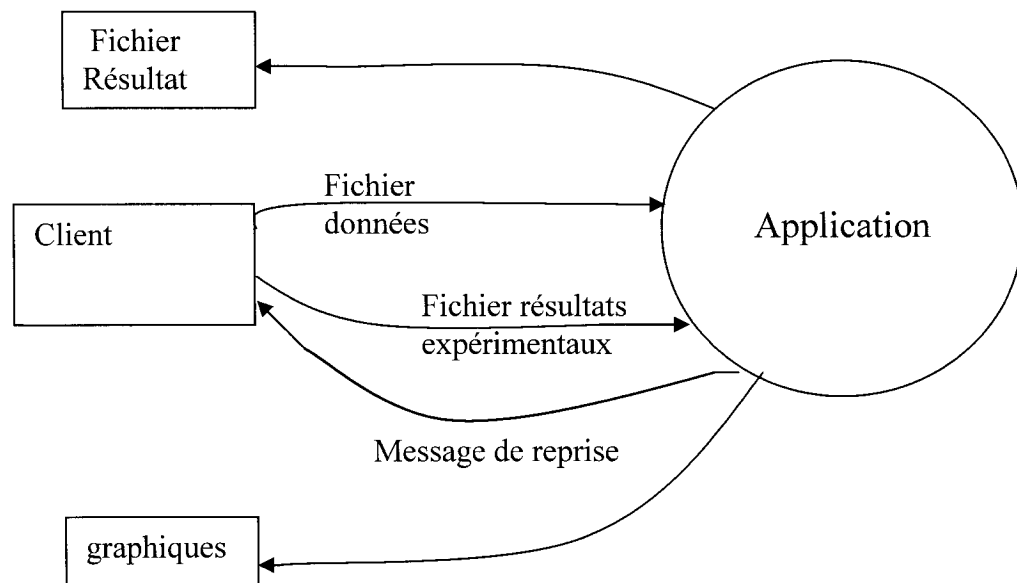


Figure 4.2 : Diagramme de contexte.

La décomposition de ce diagramme de contexte nous permet d'avoir le diagramme de flux de niveau 1, le DFD1, dont les caractéristiques sont présentées à la figure 4.3.

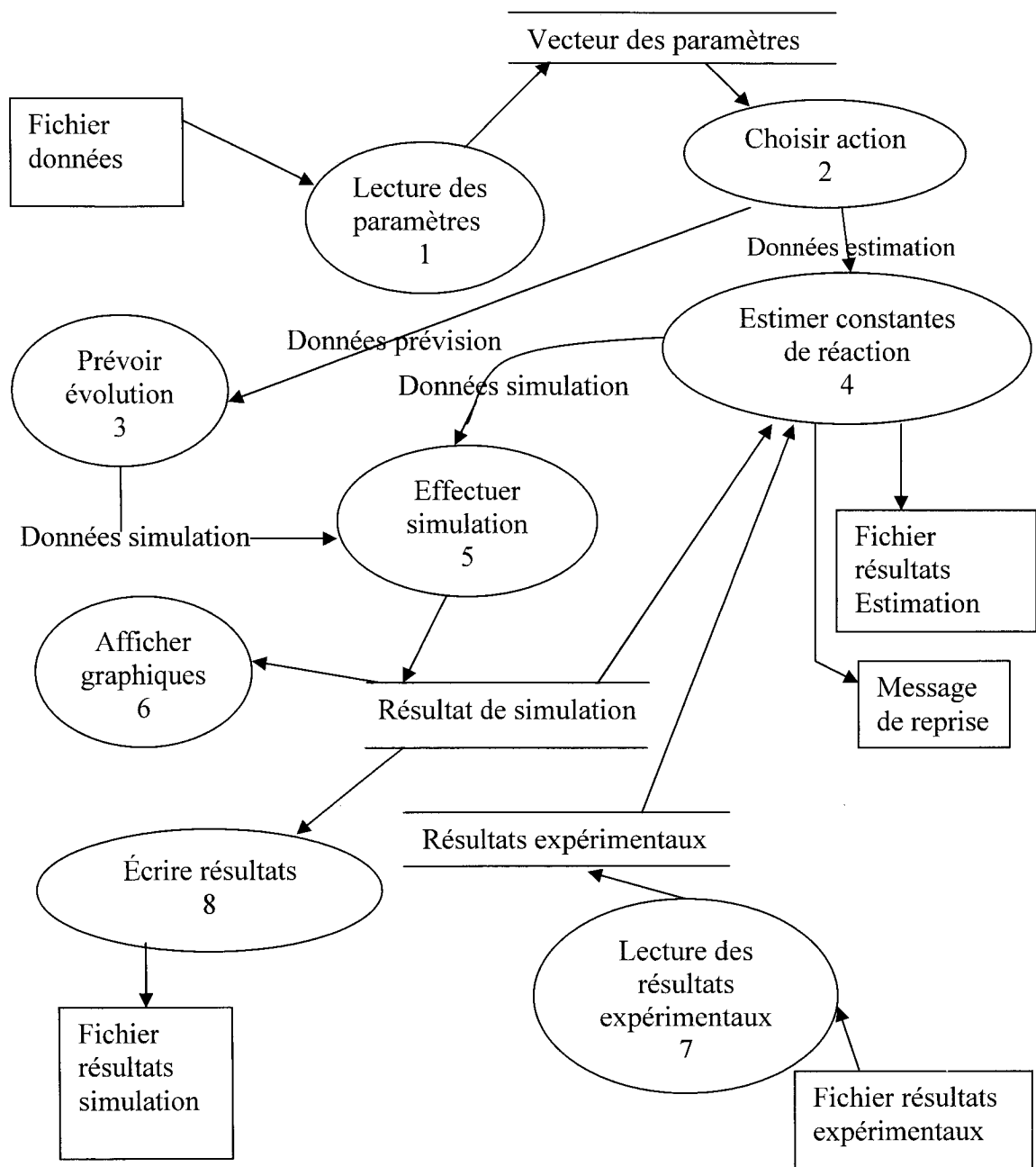


Figure 4.3 : DFD de niveau 1.

La décomposition permet donc d'obtenir des processus plus détaillés relativement aux processus numérotés dans la figure ci-dessus. Le résultat de cette analyse est d'abord

présenté aux figures 4.4 et 4.5 pour la fonction "Effectuer une simulation". Ensuite à la figure 4.6 on considère le processus "Afficher graphique", alors que le cas de l'impression des résultats est traité à la figure 4.7. Enfin, le processus "Estimer les constantes" est montré à la figure 4.8.

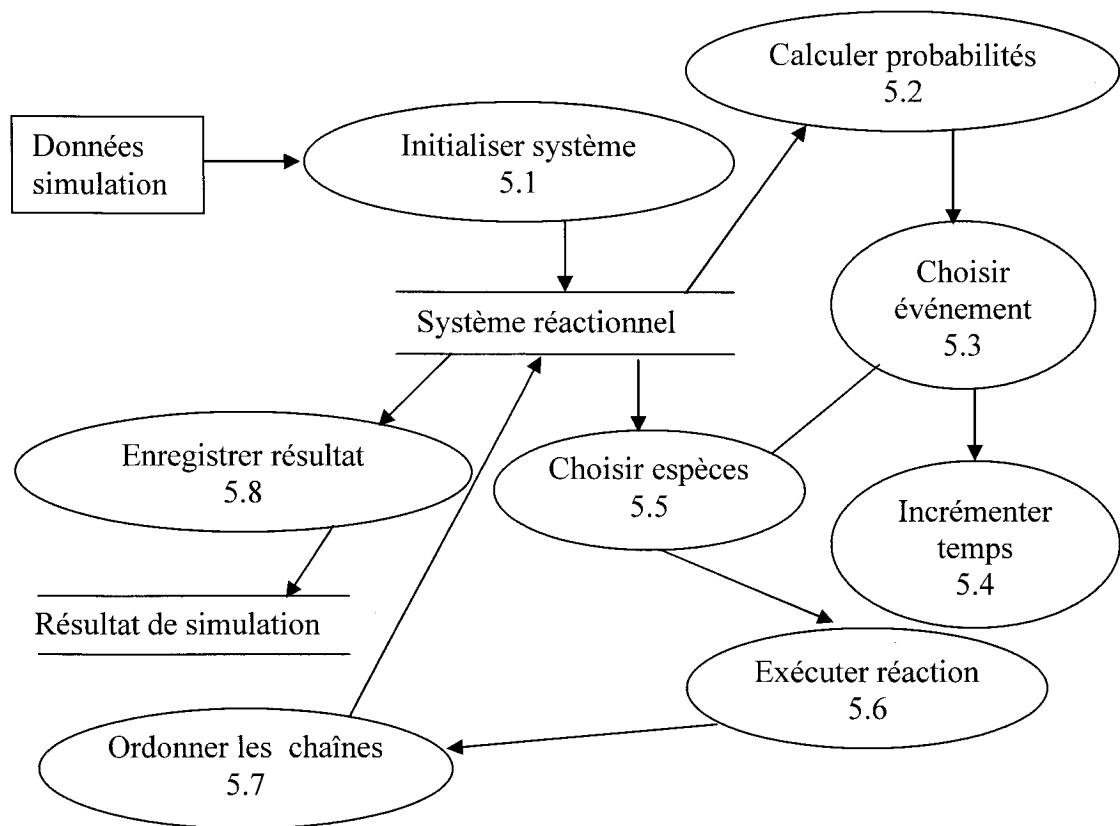


Figure 4.4 : DFD de niveau 2 de la simulation.

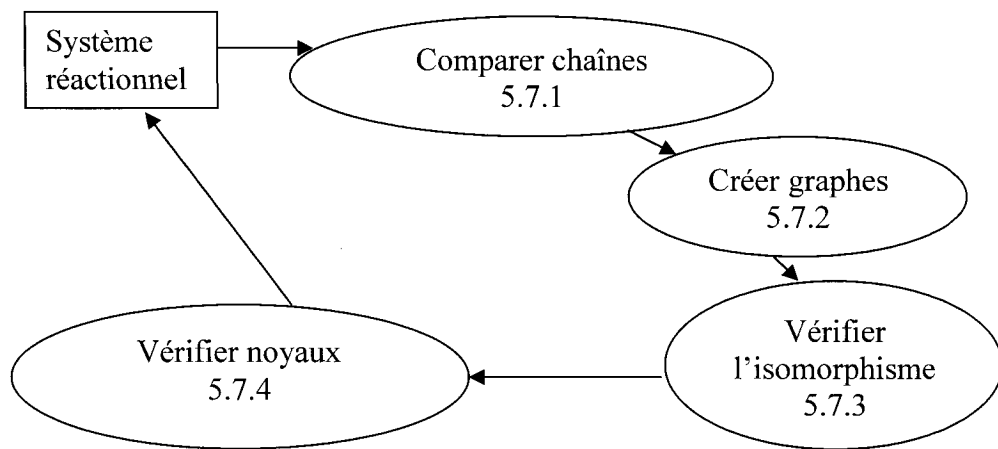


Figure 4.5 : DFD de niveau 3 de la simulation correspondant à ordonner les chaînes.

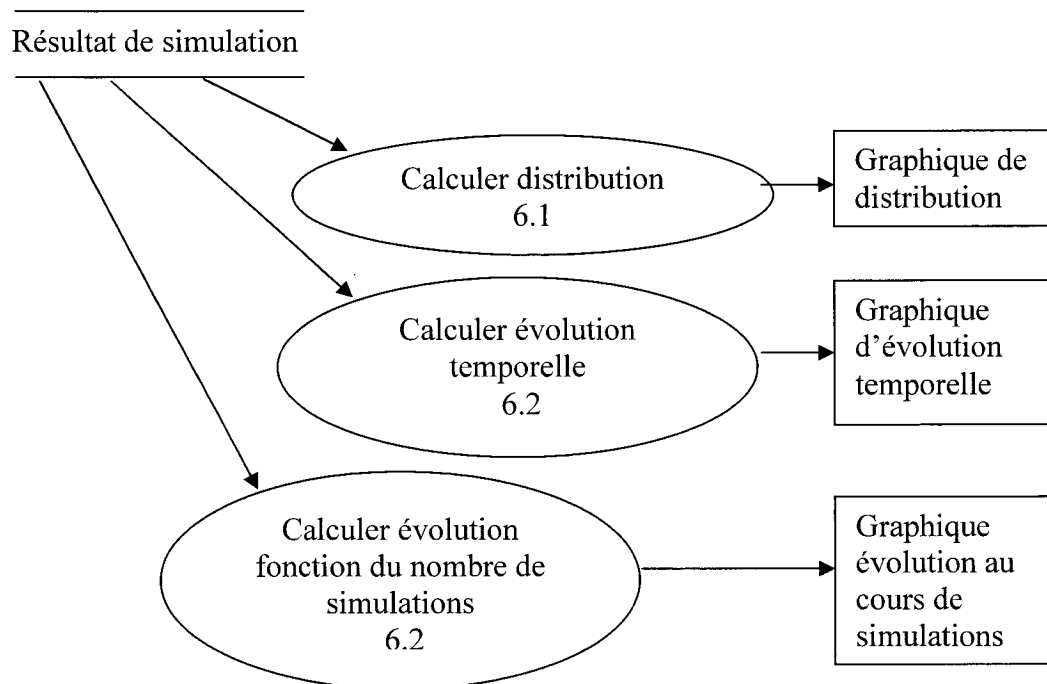


Figure 4.6 : DFD de niveau 2 d'« Afficher graphiques ».

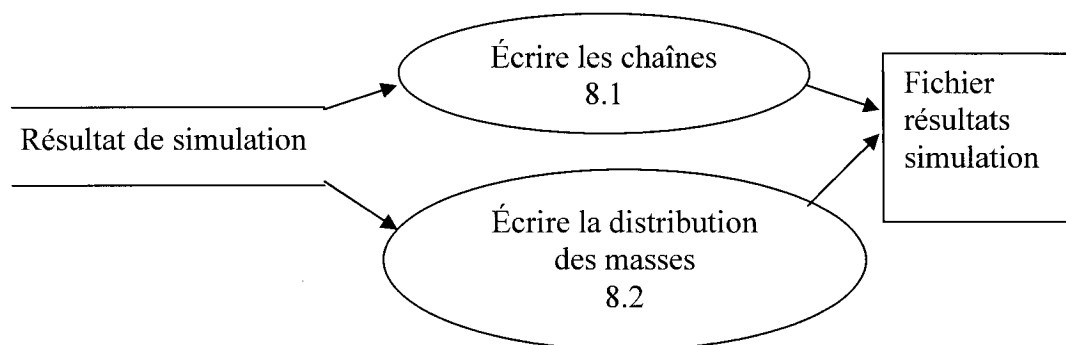


Figure 4.7 : DFD de niveau 2 d' « Écrire résultats ».

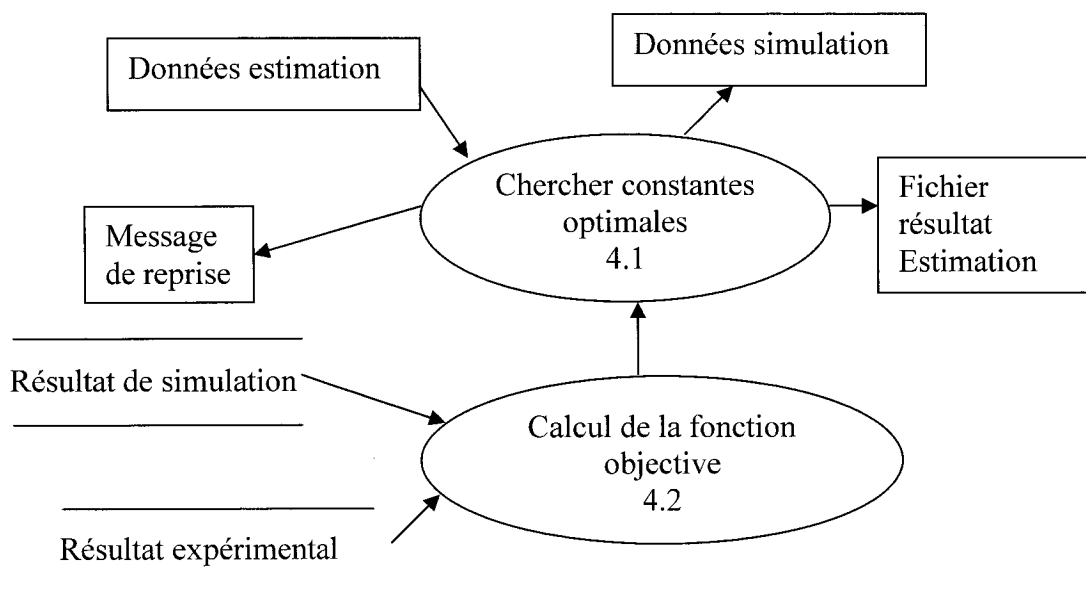


Figure 4.8 : DFD de niveau 2 d' « Estimer constantes de réaction ».

Par la méthode de décomposition des flux, nous avons identifié les principaux processus que contient l'application ainsi que les échanges des flux de données qui s'effectuent entre eux. Ceci nous permet maintenant de spécifier les algorithmes des principales fonctions composant cette application ainsi que les données et les structures de l'information qu'elle est susceptible de manipuler.

4.2 Algorithmes et dictionnaire des variables

4.2.1 Structure des données

Il ressort de l'analyse faite précédemment que les principaux flux de données et les entités de l'application pourront être représentés par les structures suivantes :

- Données estimation : représentées par les valeurs des constantes c_i , c_p , c_t , des pas de recherche dans les trois directions dc_i , dc_p , dc_t et les valeurs des quantités de monomères et du co-initiateur M_o et I_o .
- Données prévision : représentées par les valeurs des constantes c_i , c_p , c_t et les valeurs des quantités de monomères et du co-initiateur M_o et I_o .
- Vecteur des paramètres : représenté par les valeurs de l'action à entreprendre, les valeurs des constantes c_i , c_p et c_t , des pas de recherche dans les trois directions dc_i , dc_p et dc_t , les valeurs des quantités de monomères et du co-initiateur M_o et I_o , du type d'initiateur, du critère de convergence des simulations et finalement des noms de fichiers de résultats.
- Résultat expérimental : représenté par la valeur du nombre de sources d'information, la valeur du nombre des observations, d'une matrice contenant les observations et d'un tableau contenant les valeurs des instants de réalisation de ces observations.
- Résultat de simulation : représenté par une matrice contenant les chaînes formées lors des simulations, des matrices contenant les valeurs de l'évolution des probabilités des événements, d'un tableau représentant le nombre des différentes chaînes formées et d'une matrice contenant les valeurs des instants de contrôle.
- Système réactionnel : représenté par un tableau de chaînes, du nombre de chaînes différentes, du nombre de monomères et du nombre d'initiateurs.
- Coordonnée : représenté par les valeurs de l'abscisse et de l'ordonnée.

- Noyau : représenté par les valeurs du nombre d'atomes de carbone, du nombre d'atomes d'oxygène, du nombre d'atomes d'hydrogène et du nombre d'atomes d'azote.
- Liaison : représentée par les valeurs de l'indice des coordonnées, de l'état et du nombre d'unités répétitives.
- Chaîne : représentée par les valeurs du type d'initialiseur, du nombre des liaisons et de sa fréquence et d'un tableau de liaisons.
- Initialiseur : représenté par les valeurs du nombre de ses noyaux, du nombre de ses liaisons fixes et du nombre de ses liaisons variables, d'un tableau de noyaux, d'un tableau de coordonnées des liaisons fixes et d'un tableau de coordonnées des liaisons variables.
- Graphe : représenté par la valeur du nombre de noyaux, la matrice adjacente, le tableau des arcs entrant pour chaque noeud, le tableau des arcs sortant de chaque noeud, la matrice contenant pour chaque noeud les étiquettes des noeuds précédents, la matrice contenant pour chaque noeud les étiquettes des noeuds successeurs, et d'un tableau contenant les étiquettes des noeuds.
- État : représenté par la valeur du nombre de noeuds isomorphes, de deux tableaux contenant les étiquettes des noeuds isomorphes, et de deux pointeurs sur les deux graphes dont on veut vérifier l'isomorphisme.

Avec ces définitions des composantes des différentes entités et structures représentant les flux de données qu'échangent les processus de l'application, nous avons déterminé le codage des principales informations que notre application va manipuler.

4.2.2 Algorithmes :

Dans cette partie nous allons présenter les algorithmes qui constituent l'ossature de notre application. Nous allons nous limiter à présenter l'algorithme des processus principaux identifiés lors de l'analyse des besoins. Nous allons nous intéresser surtout à ceux qui sont en liaison avec la simulation stochastique de l'évolution des systèmes qui

reste le noyau de notre projet, tandis que nous n'allons pas présenter les algorithmes de lecture des données à partir d'un fichier et autres fonctionnalités similaires puisqu'ils sont omniprésents dans la littérature. Le résultat est présenté dans ce qui suit :

- **Principal :**

Début

Lecture des paramètres

Lecture des résultats expérimentaux

Initialiser les variables identifiant le système

Si action =1 alors appel de « Prévoir évolution »

Sinon appel de « Estimation des constantes »

Afficher graphiques

Fin

- **Prévoir évolution :**

Début

Lire données de prévision

Effectuer n simulations

Calculer les espérances des paramètres à étudier

Afficher les espérances

Écrire les résultats des simulations

Fin

- **Estimer constantes :**

Début

Lire données d'estimation

Lire résultats expérimentaux

Initialiser « Nombre d'itération » et « Fonction objective »

Tant que ce n'est pas le minimum de la fonction objective

Si (Nombre d'itérations < Nombre maximal)

```

    Effectuer  $n$  simulations
    Calculer les espérances des variables correspondants aux
    résultats
    Calculer la valeur de la fonction objective
    Vérifier si minimum est atteint
  Sinon
    Afficher message de reprise
    Fin de la procédure
  Fin tant que
Fin

```

- **Effectuer simulation**

```

Début
  Lire données simulation
  Tant que (critère d'arrêt est non vérifié)
    Si (critère d'enregistrement)
      Enregistrer l'état du système
    Fin si
    Calculer probabilités des événements
    Choisir un événement
    Exécuter cet événement
    Calcul du critère d'arrêt
  Fin tant que
  Stocker les résultats de la simulation
Fin

```

- **Comparer deux chaînes**

```

Début
  Si un invariant est non vérifié alors résultat =faux

```

Sinon

Former les graphes correspondants aux chaînes

Vérifier l'isomorphisme des graphes

Si isomorphisme existe

Si les liaisons correspondantes ont le même état

Résultat = vrai

Sinon résultat = faux

Sinon résultat=faux

Fin

- **Afficher graphique**

Début

Appel de la bibliothèque « pyplot »

Tracer l'évolution de l'espérance de la conversion

Tracer l'évolution de l'espérance des probabilités

Tracer l'évolution de l'espérance de M_w

Tracer l'évolution de M_n , M_w et M_z en fonction des simulations

Tracer l'espérance de la distribution des masses molaires

Fin

- **Calcul distribution moyenne**

Début

Déterminer l'intervalle commun des simulations

Calculer les moyennes sur cet intervalle

Fin

À ce stade nous avons finalisé l'analyse de notre application, nous allons aborder les modalités de son implantation dans la section suivante.

4.3 Conception de l'application

L'étape de conception est une étape importante dans le cycle de développement de logiciels. En effet, elle représente le pont entre l'analyse et l'implantation de l'application informatique du sujet de l'étude. C'est une activité qui repose sur la créativité, il y a plusieurs façons de concevoir une application, toutefois certaines règles sont appréciées par les développeurs et il est avantageux de les prendre en considération. Ces recommandations sont essentielles pour préserver la qualité de l'application à développer en termes de fiabilité et d'évolutivité. Elles suggèrent de décomposer et raffiner progressivement les processus en modules de plus en plus détaillés de façon à favoriser la cohérence des modules et un faible couplage entre eux. Nous allons exposer ici la conception logique des composantes de notre programme. Il est constitué de modules contenant des fonctions réalisant un but donné. Pour mener à bien la conception des modules de notre code nous avons utilisé la méthode du « transform mapping ». Ce choix a été dicté par le fait de la quasi absence des transactions parallèles qui induisent des chemins de traitement distinctes. Ainsi, les étapes du traitement sont :

- Lecture de l'information introduite par le client;
- L'information introduite est convertie en données internes;
- Les données internes sont transformées par les modules de transformation;
- Les données transformées sont converties en données externes;
- Les données externes sont transmises sous forme de fichiers et de graphes.

Avec cette méthodologie et en tenant en compte des critères de la cohérence et du faible couplage des modules, nous avons obtenu le schéma de conception présenté sur la figure 4.9.

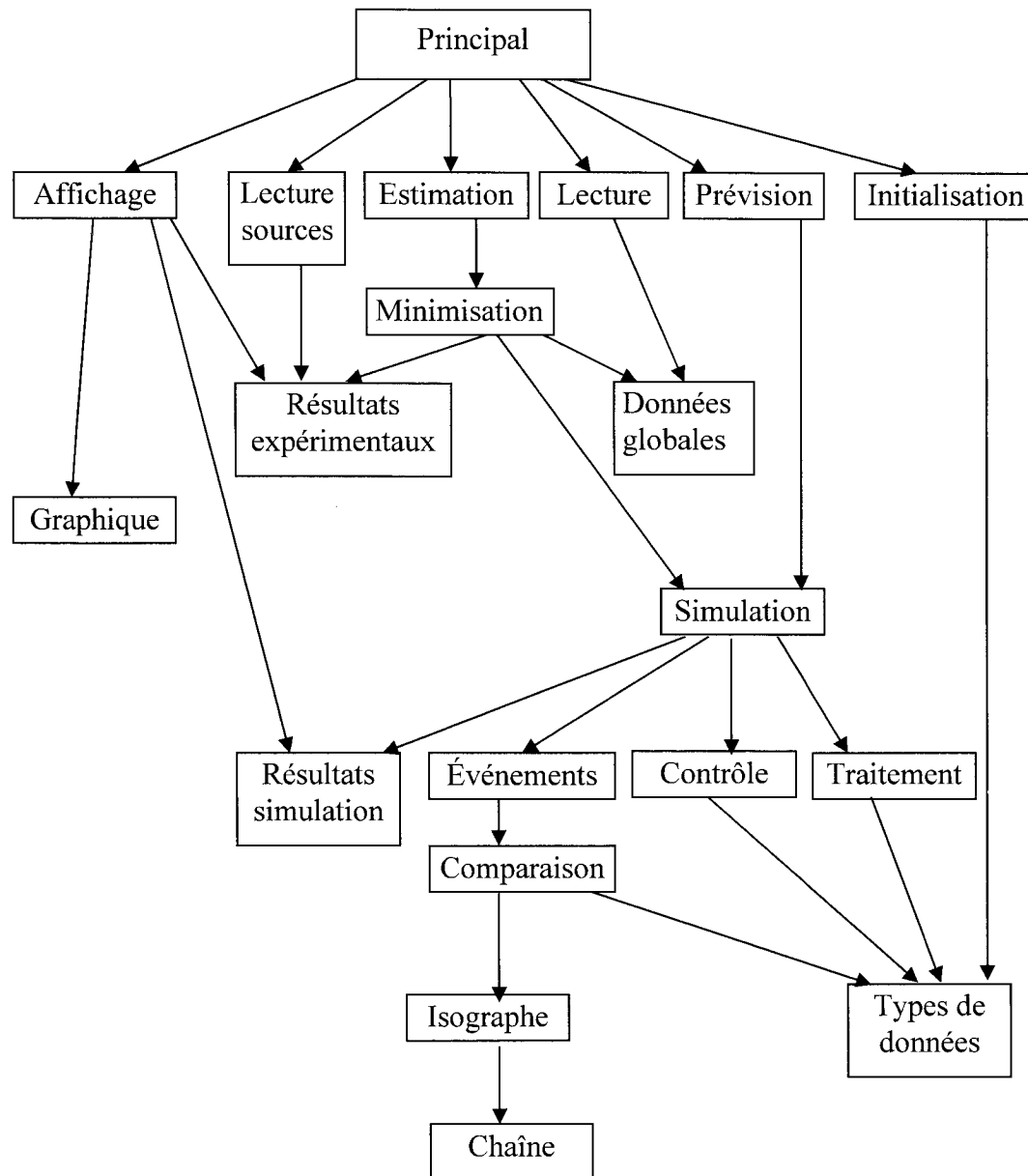


Figure 4.9 : Diagramme de conception des modules de l'application.

4.4 Application à la synthèse des polylactides

Nous nous sommes servis de notre programme pour estimer les constantes de vitesse des réactions d'initiation, de propagation et de terminaison intervenant lors de la synthèse des polylactides co-initiée par un alcool. Pour cela, nous avons utilisé les données expérimentales de l'article de Korhonen et al. (2001) pour estimer les coefficients stochastiques d'initiation c_i , de propagation c_p et de terminaison c_t . Ce choix a l'avantage d'étudier l'effet du nombre de groupements fonctionnels des co-initiateurs sur le déroulement de la polymérisation. Ainsi nous avons étudié les cas des alcools suivants : l'alcool benzylique, le 1,4-butanédiol et le pentaérythritol. La valeur du ratio M_o/I_o est fixée à 2000. Les estimations ont été faites en un premier temps en considérant la conversion comme seule source d'information puis en ajoutant, en deuxième temps, les valeurs du moment M_w comme deuxième source d'information. Les fonctions objectives f_1 et f_2 correspondants aux deux cas précités, obtenues en appliquant l'équation (3-12) sont :

$$f_1 = \sum_{i=1}^N (CV_{i,obs} - CV_{i,est})^2 \quad (4-1)$$

$$f_2 = 0.5 * \sum_{i=1}^N (CV_{i,obs} - CV_{i,est})^2 + ((Mw_{i,obs} - Mw_{i,est}) / M_2)^2 \quad (4-2)$$

Où N désigne le nombre de points observés avant l'arrêt de la synthèse;

CV la conversion;

M_w le moment d'ordre 2 de la distribution des masses molaires;

Les indices « est » et « obs » désignent les valeurs estimées et observées, respectivement.

M_2 sert à normaliser les deux membres de la somme dans le but de réaliser une pondération entre eux comme il a été mentionné dans le chapitre précédent.

Notons que la courbe d'évolution de la conversion estimée par le modèle dans le premier cas est désignée par « estimée1 » et celle du moment M_w par « estimé1 » tandis que dans le deuxième cas elles sont désignées par « estimée2 » et « estimé2 » respectivement.

En ce qui concerne le critère de convergence des résultats des simulations, il pourra être établi selon deux approches :

- la première est basée sur un nombre maximal de simulations
- la deuxième sur une valeur de précision à respecter par les trois premiers moments de la distribution des masses comme il est mentionné dans l'article de Dubois et al. (1998).

Nous avons opté pour la première approche qui nous a permis d'éviter les périodes d'échauffement nécessaires lors de la simulation stochastique des chaînes de Markov et de laisser l'initiative aux mains de l'utilisateur vu la diversité des valeurs des données expérimentales qu'il pourra mettre en jeu sans toutefois fermer la porte à l'adoption de la deuxième approche puisqu'un champ de critère de convergence est contenu dans le vecteur des paramètres de l'application. Un exemple de convergence est illustré sur la figure 4.10, d'après laquelle il apparaît clairement que la convergence est atteinte pour un nombre de simulations de l'ordre de 50 pour le cas $I_0=500$.

Mentionnons que dans le but d'étudier l'évolution des coefficients stochastiques avec le volume réactionnel, nous l'avons approché à une constante près par le nombre initial des monomères M_0 vu la valeur du rapport monomère/ initiateur qui est de l'ordre de 2000.

Il reste finalement à signaler que vu l'aspect de sauts que présente la méthode du prisme pour la recherche du minimum d'une fonction, une prospection aux alentours du point de convergence de cette méthode s'avère nécessaire pour s'assurer du résultat obtenu.

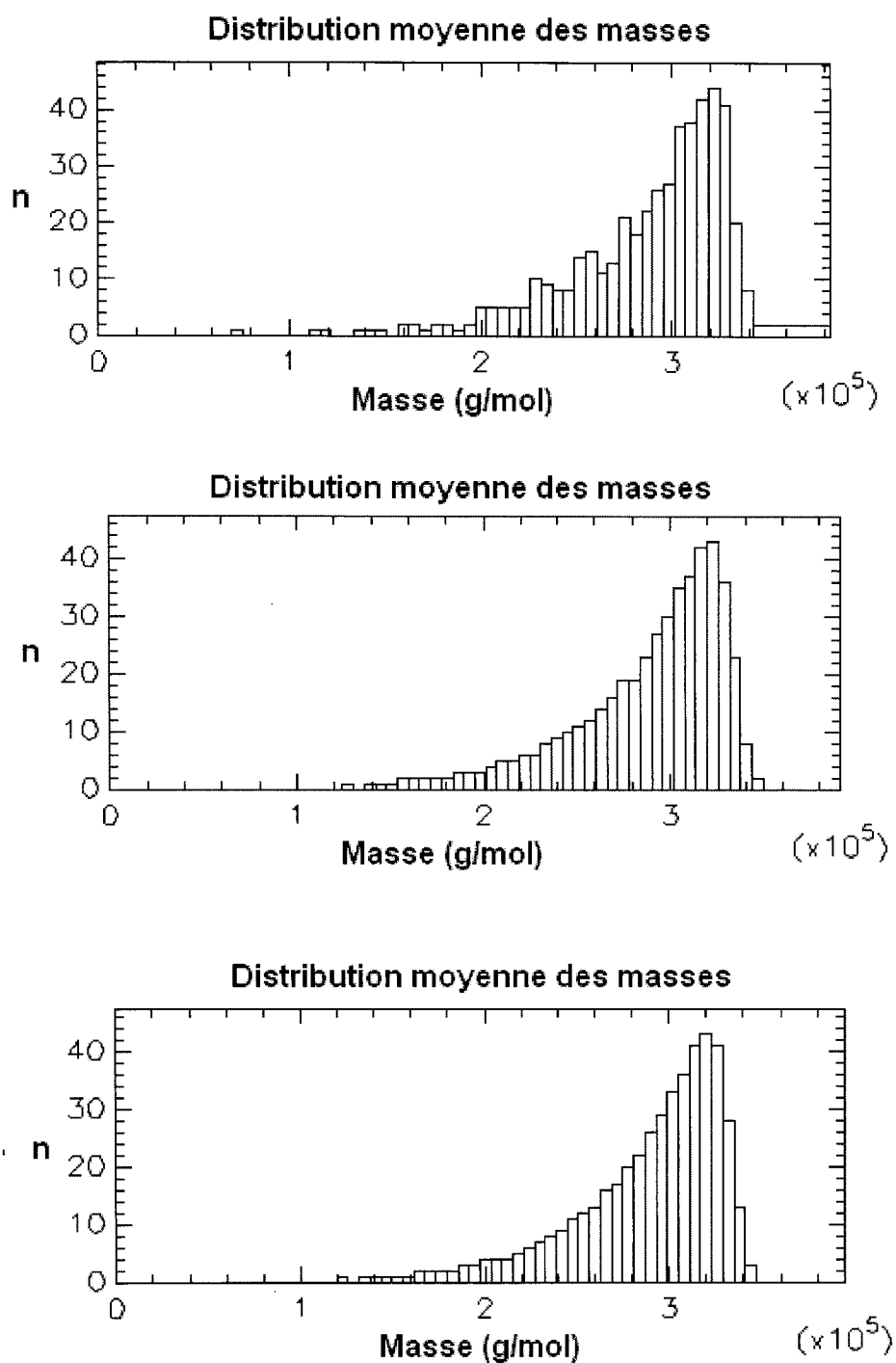


Figure 4.10 : Estimation de la distribution massique M_n dans le cas de l'alcool benzylique après 5, 50 et 100 simulations respectivement ($I_0=500$).

4.4.1 Alcool benzylique

L'alcool benzylique de formule $C_6H_5CH_2OH$ présente un seul site réactionnel (-OH).

- **Estimation des coefficients stochastiques**

La valeur de M_2 est prise égale à $5 \text{ E}5$ dans l'équation (4-2). Le choix de la valeur de M_2 est dicté par les valeurs prises par la conversion et le moment M_w . Les résultats obtenus sont présentés ci-dessous:

Tableau 4-1 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence de l'alcool benzylique avec 1 source

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_1
10000	5	6.0 E-7	2.0 E-4	1.0 E-5	6.0 E-3	2.00	1.5 E-2
100000	50	5.0 E-8	2.0 E-5	1.0 E-5	5.0 E-3	2.00	1.8 E-2
200000	100	2.4 E-8	9.9 E-6	1.1 E-5	6.0 E-3	2.48	1.8 E-2
500000	250	1.0 E-8	4.7 E-6	1.2 E-5	5.0 E-3	2.35	1.5 E-2
1000000	500	3.8 E-9	2.3 E-6	1.6 E-5	3.8 E-3	2.26	1.1 E-2
2000000	1000	2.3 E-9	1.1 E-6	1.3 E-5	4.6 E-3	2.24	1.1 E-2

Tableau 4-2 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence de l'alcool benzylique avec 2 sources

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_2
10000	5	2.7 E-6	1.6 E-4	1.0 E-5	2.7 E-2	1.60	7.9 E-2
100000	50	1.7 E-7	1.6 E-5	1.3 E-5	1.8 E-2	1.62	7.7 E-2
200000	100	6.8 E-8	9.5 E-6	1.1 E-5	1.4 E-2	1.90	10.0 E-2
500000	250	1.3 E-8	4.5 E-6	1.3 E-5	6.7 E-3	2.25	13.0 E-2
1000000	500	1.0 E-8	1.8 E-6	1.0 E-5	1.0 E-2	1.80	8.3 E-2
2000000	1000	0.6 E-8	0.8 E-6	1.1 E-5	9.6 E-3	1.64	7.7 E-2

Les valeurs des termes c_t , $c_i * M_o$ et $c_p * M_o$ présentent une faible variation avec le nombre des monomères M_o , nous déduisons que le coefficient de terminaison est quasi constant et que ceux de la propagation et de l'initiation sont inversement proportionnels à M_o . Ce fait est observé plus nettement dans le cas d'une seule source d'information, la conversion, que dans celui de deux sources d'information.

D'autre part, en ce qui concerne la concordance entre les résultats estimés et expérimentaux présentés aux figures 4.11.a et 4.11.b nous avons soulevé que :

- Le niveau de concordance obtenu avec les données de la conversion est meilleur que celui obtenu avec les données du moment M_w ;
- L'ajout de la deuxième source d'information améliore la concordance des données du moment M_w et altère légèrement celle des données de la conversion;
- Les divergences entre les données expérimentales et estimées sont plus grandes au début et vers la fin de la réaction de synthèse;
- Les courbes estimées sont plus lisses que les courbes expérimentales.

Nous déduisons que nous avons pu approcher l'évolution de notre système étudié et réduire les différences entre les courbes des M_w , expérimentale et estimées par l'ajout d'une deuxième source d'information sans toutefois résoudre définitivement les anomalies observées au début et vers la fin de la réaction de synthèse. Ces anomalies sont dues principalement au temps de mélange que requiert un système réactionnel pour être considéré comme un mélange parfait et aux réactions de dégradation qui apparaissent vers la fin de la réaction. Néanmoins, les résultats que nous avons obtenus sont satisfaisants dans l'ensemble.

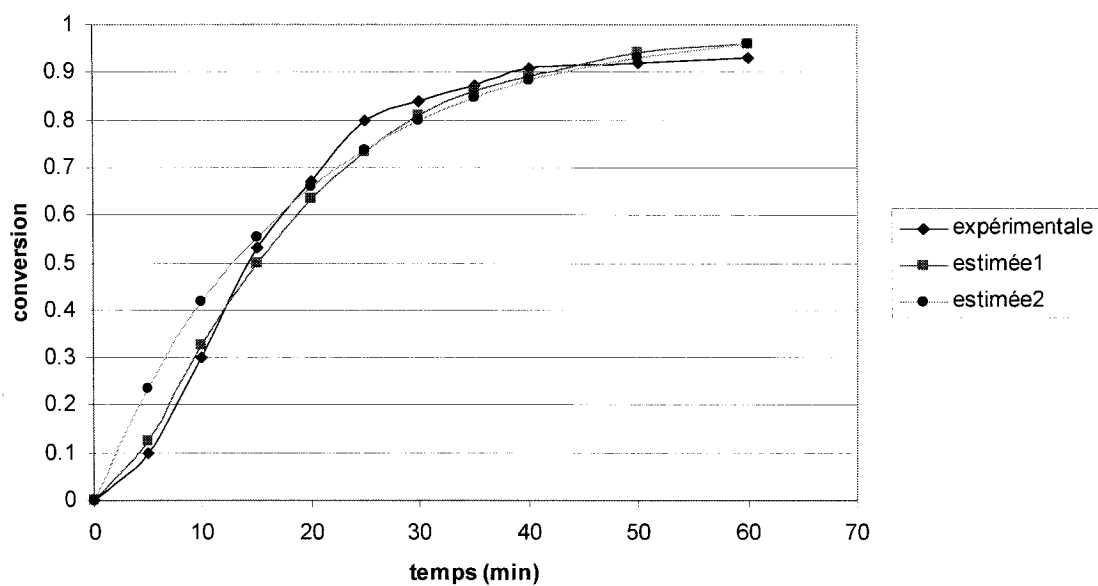


Figure 4.11.a: Comparaison de la conversion expérimentale et estimée pour la synthèse des polylactides co-initiée par l'alcool benzylique.

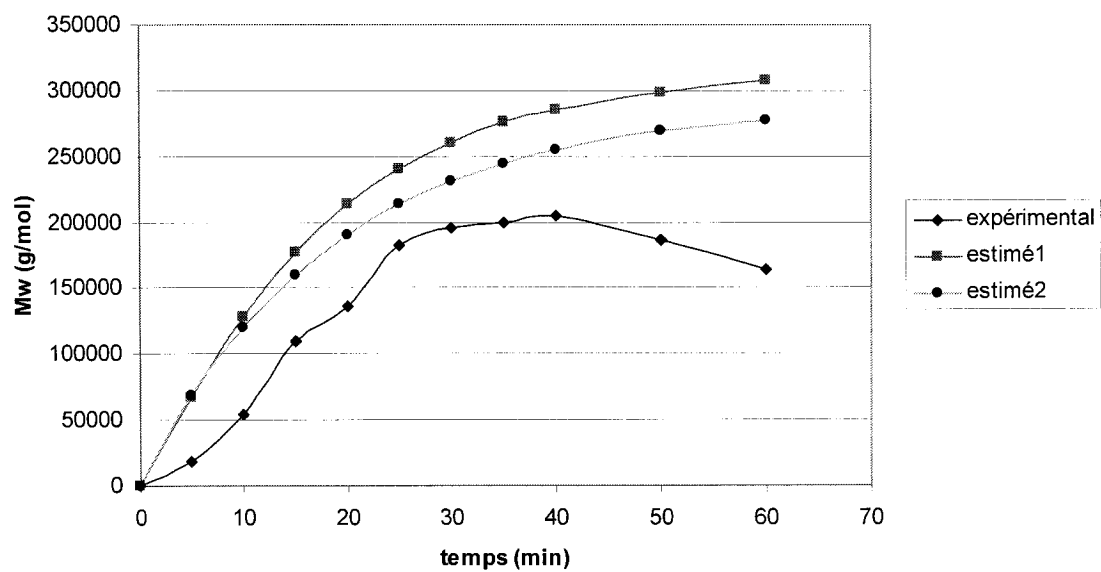


Figure 4.11.b: Comparaison de M_w expérimental et estimé pour la synthèse des polylactides co-initiée par l'alcool benzylique.

- **Détermination des constantes de réaction k_i , k_p et k_t :**

Pour estimer les constantes de réactions nous avons utilisé les équations (3-4), (3-5) et (3-6). Ainsi k_t est prise égale à la moyenne des valeurs obtenues pour le coefficient stochastique c_t , k_p est prise égale à la pente de la droite de régression des valeurs obtenues des coefficients c_p et k_i est prise égale à la pente de la droite de régression des valeurs estimées des coefficients stochastiques c_i . Les résultats sont présentés dans le tableau suivant :

Tableau 4-3 : Valeurs des constantes de réactions de la synthèse des polylactides en présence de l'alcool benzylique

Cas étudié	k_i (1/s M_o)	k_p (1/s M_o)	k_t (1/s)
Avec 1 source	5.99 E-3	20 E-1	1.20 E-5
Avec 2 sources	2.69 E-3	16 E-1	1.13 E-5

- **Estimation de l'évolution du système :**

Nous avons simulé l'évolution du système pour obtenir l'estimation de la distribution massique M_n et des probabilités d'occurrence des réactions en utilisant les constantes de vitesse des réactions estimées avec deux sources d'information. D'après la courbe de la distribution massique illustrée sur la figure 4.12.a obtenue pour un échantillon de $I_o = 500$ et $M_o = 1 \text{ e}6$, il apparaît que celle-ci présente de l'asymétrie à gauche. Nous déduisons qu'il y a formation de plus de chaînes de masse molaire inférieure à la masse molaire moyenne. D'autre part en regardant le graphique de la figure 4.12.b correspondant aux probabilités d'occurrence des événements dans le système, nous pouvons remarquer une initialisation rapide des sites réactionnels et qui se termine dans les premières minutes, une faible occurrence des réactions de terminaison et une tendance au repos du système après 1 heure, i.e. le nombre de réactions occurrentes dans le système est négligeable. Ces constatations nous permettent d'avoir une idée sur le déroulement de la réaction de synthèse étudiée et des paramètres qui sont influents au fil du temps.

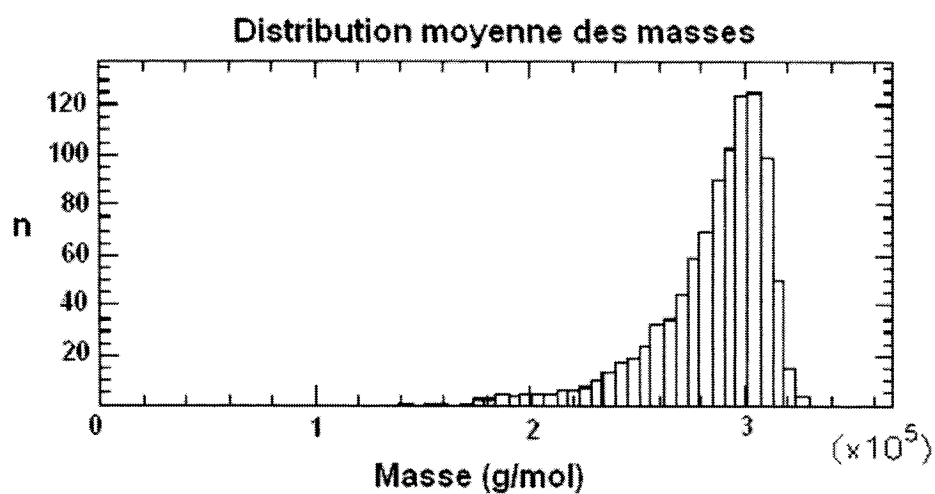


Figure 4.12.a : Estimation de la distribution massique lors de la synthèse des polylactides co-initiée par l'alcool benzylique.

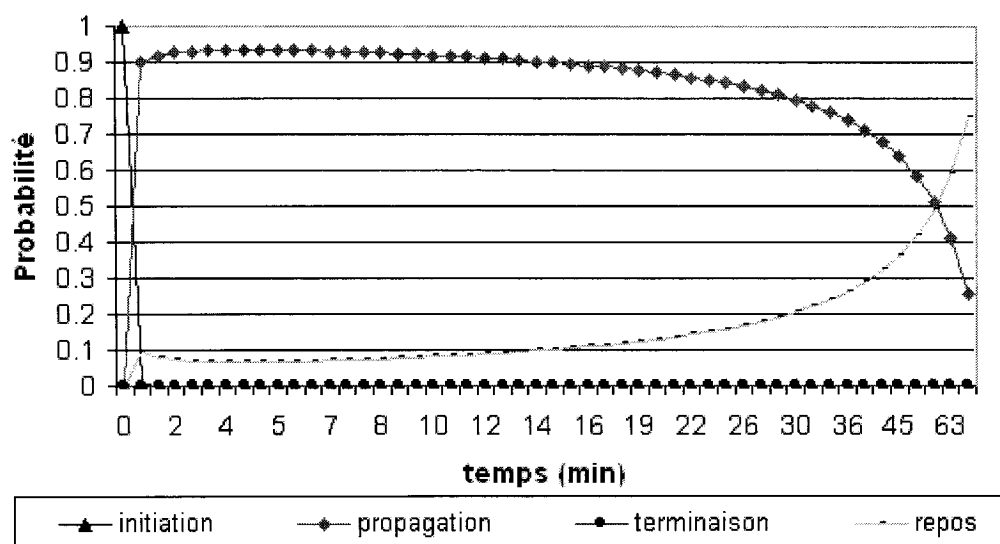


Figure 4.12.b : Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par l'alcool benzylique.

4.4.2 1,4-Butanédiol

Le 1,4-butanédiol de formule $HO(CH_2)_4OH$ présente deux sites réactionnels (-OH), les résultats obtenus sont présentés ci-dessous :

- **Valeurs des coefficients stochastiques:**

En prenant M_2 égale à $7 E5$ dans l'équation (4-2) et pour différentes valeurs de M_o et I_o nous avons obtenu les résultats présentés ci-après:

Tableau 4-4 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du 1,4-butanédiol avec 1 source

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_1
10000	5	4.25 E-7	1.80 E-4	1.2 E-5	4.2 E-3	1.80	3.0 E-2
100000	50	4.73 E-8	1.84 E-5	1.1 E-5	4.7 E-3	1.84	2.9 E-2
200000	100	2.29 E-8	7.25 E-6	1.3 E-5	4.6 E-3	1.45	3.0 E-2
500000	250	9.13 E-9	3.77 E-6	1.0 E-5	4.6 E-3	1.89	2.7 E-2
1000000	500	4.79 E-9	1.85 E-6	1.1 E-5	4.8 E-3	1.85	2.9 E-2
2000000	1000	2.80 E-9	8.62 E-6	1.3 E-5	5.6 E-3	1.72	3.1 E-2

Tableau 4-5 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du 1,4-butanédiol avec 2 sources

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_2
10000	5	4.25 E-7	1.65 E-4	1.2 E-5	4.2 E-3	1.65	3.34 E-2
100000	50	4.77 E-8	1.73 E-5	1.3 E-5	4.8 E-3	1.73	3.63 E-2
200000	100	1.89 E-8	7.20 E-6	1.3 E-5	3.8 E-3	1.44	3.54 E-2
500000	250	1.09 E-8	3.25 E-6	1.3 E-5	5.4 E-3	1.63	3.55 E-2
1000000	500	5.30 E-9	1.78 E-6	1.2 E-5	5.3 E-3	1.78	3.55 E-2
2000000	1000	4.5 E-9	7.5 E-7	1.2 E-5	9.0 E-3	1.50	3.64 E-2

Les valeurs des termes c_i , $c_i * M_o$ et $c_p * M_o$ montrent que l'évolution des coefficients stochastiques suit de nouveau la règle stipulant que le coefficient de

terminaison est presque constant et que ceux de la propagation et de l'initiation sont inversement proportionnels à M_0 . Cette règle est plus respectée dans le cas d'une seule source d'information, la conversion, que dans celui de deux sources d'information.

D'autre part, en ce qui concerne la concordance entre les résultats estimés et expérimentaux illustrés aux figures 4.13.a et 4.13.b nous avons soulevé que :

- Le niveau de concordance entre les données expérimentales et estimées de la conversion est meilleur que celui entre les données du moment M_w ;
- L'ajout de la deuxième source d'information améliore légèrement la concordance des données expérimentales et estimées du moment M_w ainsi que celle des données de la conversion;
- Les divergences entre les données expérimentales et estimées sont plus grandes au début et vers la fin de la réaction de synthèse;
- Les courbes expérimentales présentent un aspect sigmoïdal, notamment celle de la conversion.
- Le niveau de concordance entre les données expérimentales et estimées a subi une régression par comparaison à celui obtenu pour l'alcool benzylique.

Malgré le fait que nous avons pu réduire les différences entre les deux courbes des M_w , expérimentale et estimée par l'ajout d'une deuxième source d'information nous n'avons pas pu atteindre le niveau de concordance obtenu dans le cas de l'alcool benzylique. Cette constatation nous amène en plus des anomalies dues aux temps requis pour réaliser le mélange parfait et des réactions de dégradation à reconsidérer l'hypothèse de l'équiréactivité des sites réactionnels. La réactivité des sites réactionnels est influencée parmi d'autres par la longueur des branches de la chaîne.

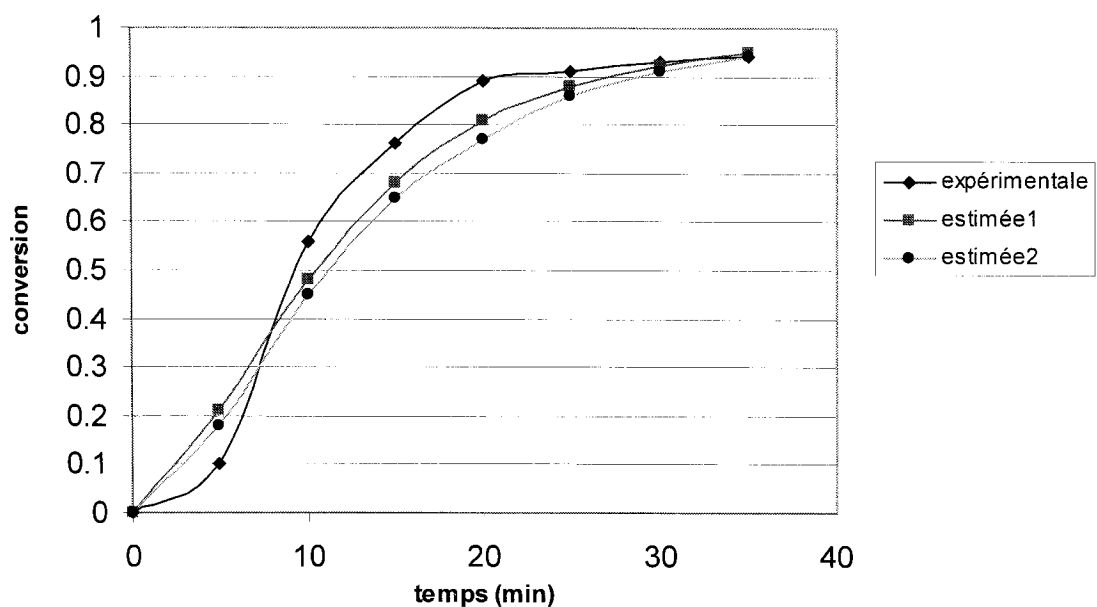


Figure 4.13.a: Comparaison de la conversion expérimentale et estimée pour la synthèse des polylactides co-initiée par le 1,4-butanédiol.

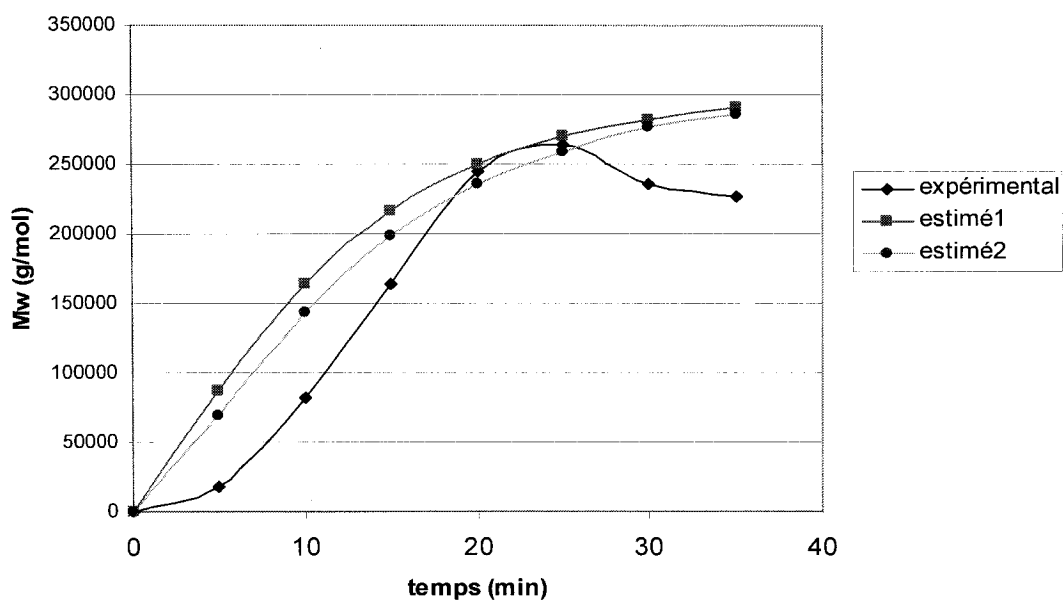


Figure 4.13.b: Comparaison du M_w expérimental et estimé pour la synthèse des polylactides co-initiée par le 1,4-butanédiol.

Détermination des constantes de réaction k_i , k_p et k_t :

De la même façon que celle mentionnée pour le cas de l'alcool benzylique, nous avons obtenu les résultats pour les constantes de vitesse de réaction présentés dans le tableau ci-dessous :

Tableau 4-6 : Valeurs des constantes de réactions de la synthèse des polylactides en présence du 1,4-butanediol

Cas étudié	k_i (1/s Mo)	k_p (1/s Mo)	k_t (1/s)
Avec 1 source	4.26 E-3	1.80	1.19 E-5
Avec 2 sources	4.25 E-3	1.65	1.17 E-5

Les constantes de vitesse ne diffèrent pas beaucoup dans les deux cas, notamment en ce qui concerne les réactions d'initiation et de terminaison. Ce fait est lié à la non amélioration de la concordance des courbes expérimentales et estimées.

- **Estimation de l'évolution du système:**

Une fois les constantes de vitesse des réactions estimées pour le cas de deux sources d'information, nous avons simulé l'évolution du système pour obtenir l'estimation de la distribution massique M_n et des probabilités d'occurrence des réactions comme il est illustré sur les figures 4.14.a et 4.14.b ci-après. Il apparaît que la distribution massique obtenue pour un échantillon de $I_o = 500$ et $M_o = 1 \text{ e}6$, présente de l'asymétrie à gauche ainsi qu'une légère bimodalité. Elle est plus large comparée à celle du cas de l'alcool benzylique. D'autre part en regardant le graphique correspondant aux probabilités d'occurrence des événements dans le système, nous pouvons constater une initialisation rapide des sites réactionnels, une opération qui se termine dans les premières minutes, une faible occurrence des réactions de terminaison et une tendance au repos du système après 35 minutes.

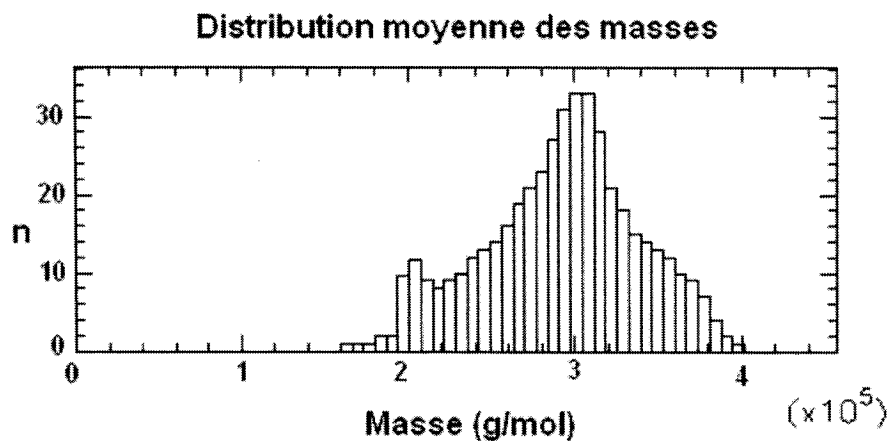


Figure 4.14.a : Estimation de la distribution massique lors de la synthèse des polylactides co-initiée par le 1,4-butanédiol

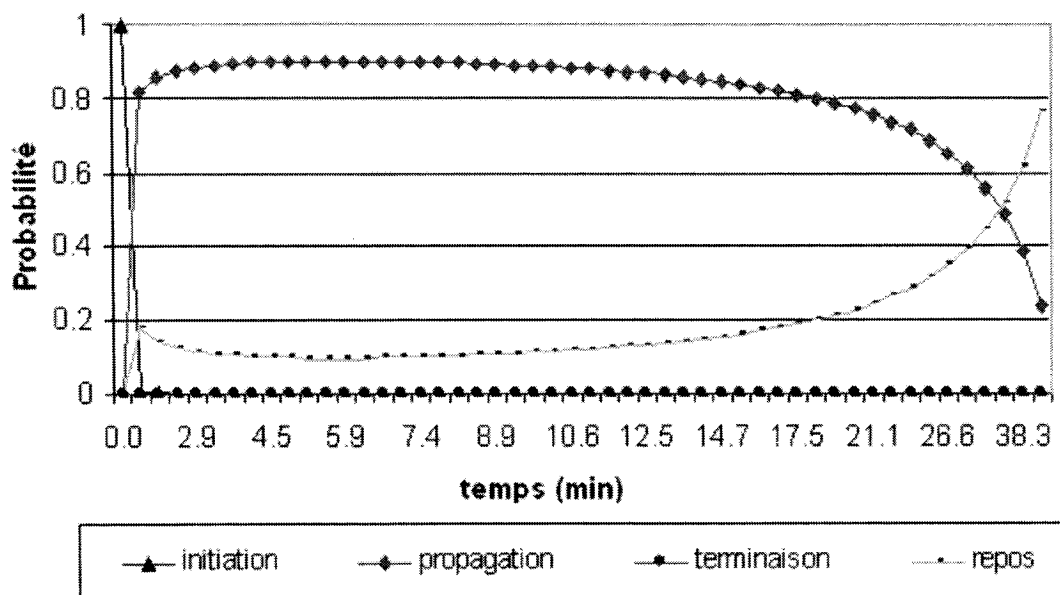


Figure 4.14.b : Estimation de l'évolution des événements lors de la synthèse des polylactides co-initiée par le 1,4-butanédiol.

4.4.3 Pentaérythritol :

Le pentaérythritol de formule $C(CH_2OH)_4$ présente quatre sites réactionnels, les résultats que nous avons obtenus sont présentés dans ce qui suit.

- **Valeurs des coefficients stochastiques:**

En prenant M_2 égale à 5 E5 dans l'équation (4-2) et pour différentes valeurs de M_o et I_o nous avons obtenu les résultats présentés dans les tableaux ci-dessous :

Tableau 4-7 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du pentaérythritol avec 1 source

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_1
10000	5	3.7 E-7	1.34 E-4	1.1 E-5	3.7 E-3	1.34	5.17 E-2
100000	50	5.6 E-8	1.22 E -5	1.0 E-5	5.6 E-3	1.22	5.63 E-2
200000	100	2.0 E-8	7.03 E -6	1.2 E-5	4.1 E-3	1.41	5.12 E-2
500000	250	1.1 E-8	3.77 E-6	1.0 E-5	5.5 E-3	1.89	5.36 E-2
1000000	500	2.6 E-9	1.69 E-6	1.1 E-5	2.6 E-3	1.69	4.76 E-2
2000000	1000	5.0E-10	8.62E-7	1.1 E-5	1.0 E-3	1.72	5.22 E-2

Tableau 4-8 : Résultats de l'estimation des coefficients stochastiques lors de la synthèse des polylactides en présence du pentaérythritol avec 2 sources

M_o	I_o	c_i	c_p	c_t	$c_i * M_o$	$c_p * M_o$	f_2
10000	5	4.75 E-7	1.21 E-4	1.4 E -5	4.7 E-3	1.21	2.88 E-2
100000	50	5.03 E-8	1.19 E -5	1.7 E -5	5.0 E-3	1.19	2.93 E-2
200000	100	3.43 E-8	5.92 E -6	1.7 E -5	6.8 E-3	1.18	3.18 E-2
500000	250	1.66 E-8	2.50 E -6	1.1 E -5	8.3 E-3	1.25	3.55 E-2
1000000	500	4.15 E-9	1.35 E-6	1.2 E-5	4.1 E-3	1.35	2.63 E-2
2000000	1000	2.2 E-9	7.5 E-7	1.2 E-5	4.4 E-3	1.50	2.84 E-2

Les valeurs de c_t , $c_i * M_o$ et $c_p * M_o$ montrent que le coefficient de terminaison est presque constant et que ceux de la propagation et de l'initiation sont inversement proportionnels à M_o . Cette règle est moins respectée que dans les cas de l'alcool benzylique et du 1,4 butanédiol.

D'autre part, en ce qui concerne la concordance entre les résultats estimés et expérimentaux présentés sur les figures 4.15.a et 4.15.b nous avons soulevé que :

- Les niveaux de concordance entre les données expérimentales et estimées de la conversion du moment M_w sont identiques;
- L'ajout de la deuxième source d'information améliore légèrement la concordance des données expérimentales et estimées du moment M_w ainsi que celle des données de la conversion;
- Les divergences entre les données expérimentales et estimées sont plus grandes au début de la réaction de synthèse;
- Les courbes expérimentales de la conversion et du M_w présentent un aspect sigmoïdal;
- Le niveau de concordance entre les données expérimentales et estimées a subi une régression par comparaison à celui obtenu pour l'alcool benzylique et du 1,4 butanédiol.

Le degré de concordance entre les courbes expérimentales et estimées et l'aspect sigmoïdal des courbes expérimentales de la conversion et du M_w nous permet d'avancer que nous sommes face à une limitation du modèle adopté. En effet, les données expérimentales présentent deux points d'inflexion qui correspondent aux changements dans l'évolution du système dû au temps requis pour réussir le mélange des réactifs et à la présence de réactions intramoléculaires et intermoléculaires secondaires alors que les courbes estimées garde une allure croissante.

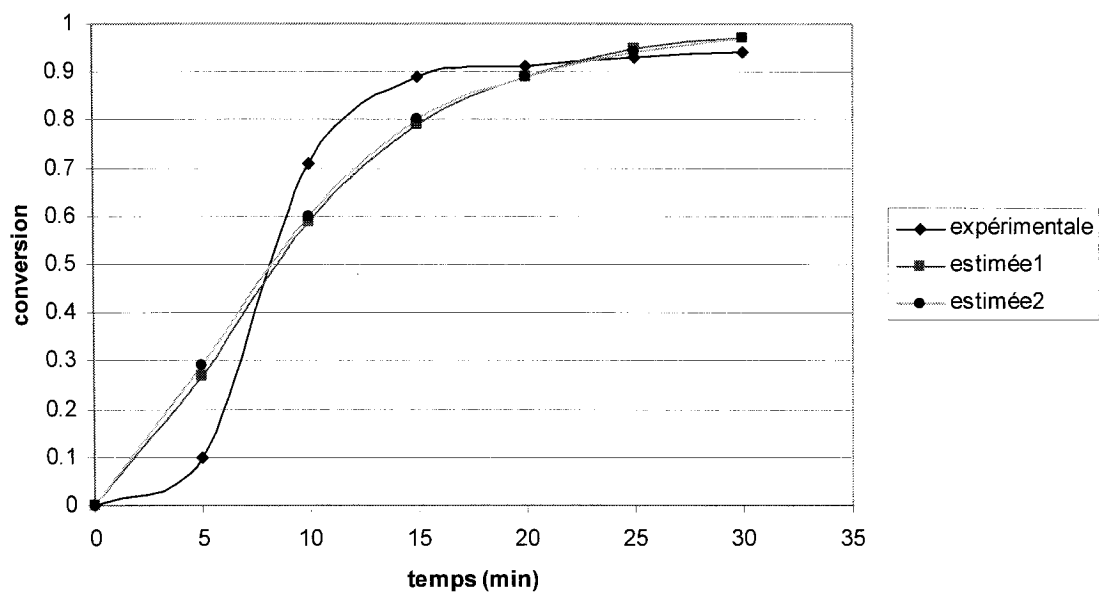


Figure 4.15.a: Comparaison de la conversion expérimentale et estimée pour la synthèse des polyactides co-initiée par le pentaérythritol

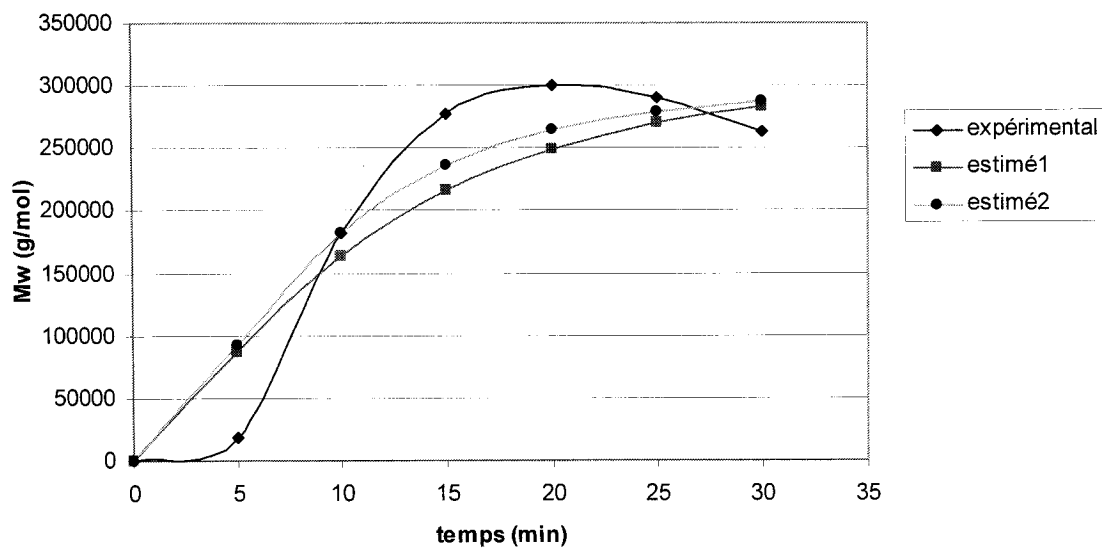


Figure 4.15.b: Comparaison du Mw expérimental et estimé pour la synthèse des polyactides co-initiée par le pentaérythritol.

- **Détermination des constantes de réaction k_i , k_p et k_t :**

Par la même méthodologie que celle mentionnée pour le cas de l'alcool benzylique, nous avons obtenu les résultats présentés dans le tableau ci-dessous :

Tableau 4-9 : Valeurs des constantes de réactions de la synthèse des polylactides co-initiée par le pentaérythritol

Cas étudié	k_i (1/s Mo)	k_p (1/s Mo)	k_t (1/s)
Avec 1 source	3.75 E-3	1.34	1.22 E-5
Avec 2 sources	4.77 E-3	1.21	1.33 E-5

- **Estimation de l'évolution du système :**

De la même façon que pour les deux alcools traités auparavant, nous avons simulé l'évolution du système pour obtenir l'estimation de la distribution massique M_n et des probabilités d'occurrence des réactions. Il apparaît donc que la distribution obtenue pour un échantillon de $I_o = 500$ et $M_o = 1 \text{ e}6$ présente de l'asymétrie à droite ainsi qu'une légère bimodalité comme il est présenté par la figure 4.16.a. D'autre part en regardant le graphique de la figure 4.16.b correspondant aux probabilités d'occurrence des événements dans le système, nous pouvons constater une initialisation rapide des sites réactionnels, une opération qui se termine dans les premières minutes, une faible occurrence des réactions de terminaison et une tendance au repos du système après 25 minutes.

Il apparaît que le temps de terminaison de la réaction de synthèse des polylactides diminue avec le nombre de groupements fonctionnels de l'initiateur, passant de plus qu'une heure pour l'alcool benzylique à 38 minutes pour le butanédiol et 25 minutes pour le pentaérythritol. Ainsi, plus nous avons de points de croissance par chaîne plus la conversion des monomères est rapide.

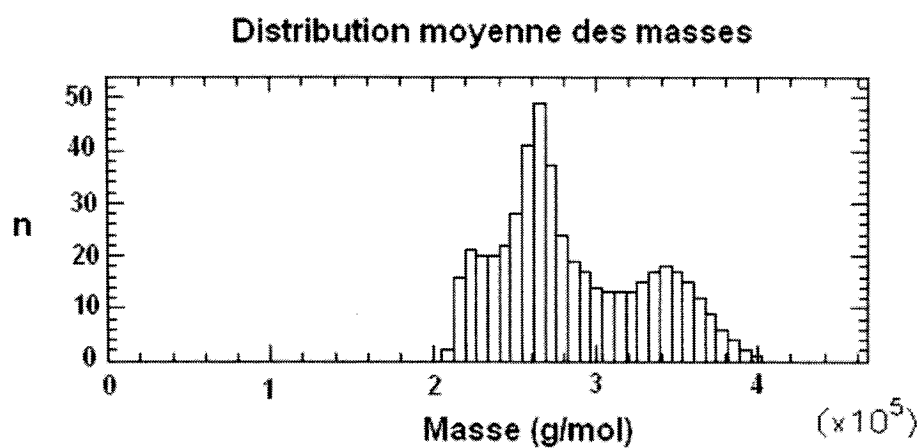


Figure 4.16.a : Estimation de la distribution massique lors de la synthèse des polyactides co-initiée par le pentaérythritol.

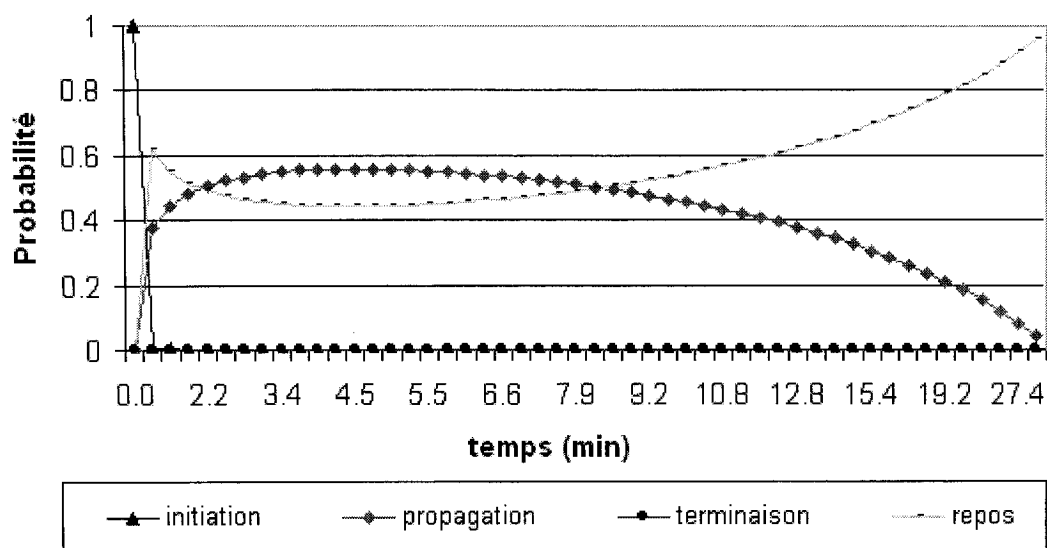


Figure 4.16.b : Estimation de l'évolution des événements lors de la synthèse des polyactides co-initiée par le pentaérythritol.

4.5 Interprétation

Le code donne des résultats satisfaisants dans l'ensemble. Nous pouvons constater que les constantes de vitesse des réactions que nous avons obtenues sont assez voisines entre elles et que les résultats générés avec deux sources d'information sont plus plausibles que ceux obtenus avec une seule source. Cependant, il faut prendre en compte que pour réaliser de bonnes estimations il faut avoir assez de données, le nombre des observations joue un rôle important quant à la fiabilité et la qualité des estimations. Dans notre cas nous avons utilisé des observations réalisées à des intervalles de temps égaux pour les trois réactions de synthèse sans égard à la rapidité des réactions de polymérisation dans le cas d'alcools multifonctionnels, ce qui a réduit le nombre de points et par la suite altéré la qualité des estimations.

Les résultats que nous avons obtenus nous permettent de déduire que le modèle représente mieux la polymérisation des polylactides co-initiée par un initiateur monofonctionnel que celle co-initiée par un initiateur multifonctionnel ce qui peut être expliqué par la remise en cause de l'hypothèse de l'équiréactivité des sites réactionnels dans le cas d'un initiateur multifonctionnel, une hypothèse que plusieurs auteurs ont mis en doute, la prise en compte de la non équiréactivité des sites réactionnels pourra être l'objet de nouveaux travaux dans le futur. Ajoutons à ce qu'il a été dit le fait que nous avons simplifié autant que possible notre approche en adoptant un nombre minimal d'événements se produisant dans un système chimique tel la polymérisation des polylactides, en réalité plusieurs événements se produisent au cours de l'évolution d'un tel système, l'ajout d'autres événements tel la transestérification pourra améliorer d'avantage notre approche.

En ce qui concerne l'aspect bimodal observé pour le cas de l'alcool tétrafonctionnel, nous pouvons l'expliquer par le fait qu'il y a une subdivision dans la population des initiateurs. En effet, plus les nombres de chaînes à trois sites actifs, deux sites actifs et un seul site actif sont représentatifs plus nous aurons de modes dans la distribution de la masse molaire moyenne. Le système évolue comme si nous y avons mis des initiateurs de fonctionnalités différentes. Pour remédier à ce problème il faut

s'assurer qu'un bon nombre de chaînes ont la totalité de leurs sites initialisés. Ceci est obtenu par une augmentation de la constante de vitesse de l'initialisation. Pour illustrer notre idée nous avons mené des simulations avec $I_o = 500$, $M_o = 1 \text{ e}6$ et $k_i = 3.75 \text{ e-}3$ et $2.06 \text{ e-}2$ respectivement, Les courbes de la distribution de la masse molaire moyenne obtenues pour ces deux cas sont présentées sur les figures 4.17.a et 4.17.b. La forme de ces courbes confirme notre explication. Ainsi plus nous augmentons la valeur de k_i plus nous diminuons les chances d'apparition de modes dans la distribution de la masse molaire moyenne. Toutefois, nous mentionnons que l'aspect de la bimodalité des courbes de pour le cas d'initiateurs multifonctionnels pourra être l'objet de travaux ultérieurs.

Un autre fait s'ajoute qui est le caractère sigmoïdale des courbes expérimentales, cet aspect s'observe nettement pour le cas multifonctionnel que pour le cas monofonctionnel. Ainsi nous avons deux points d'inflexion au début et vers la fin des réactions, nous pouvons parler de régimes transitoires et régime permanent pour l'évolution du système et par la suite que notre modèle s'adapte mieux à expliquer le régime permanent séparant ces deux phases de l'évolution du système.

Néanmoins, en dépit de ce qui a été mentionné, la mise en œuvre de notre application constitue un outil dans la modélisation et la simulation stochastique de la polymérisation par ouverture de cycle. Ainsi, en utilisant notre code, l'utilisateur pourra prédire l'évolution de son système réactionnel et d'avoir une première prévision sur les résultats escomptés. Dans le but de la rendre plus générale, plus attractive, d'autres ajouts et améliorations sont nécessaires comme nous le verrons dans les chapitres suivants.

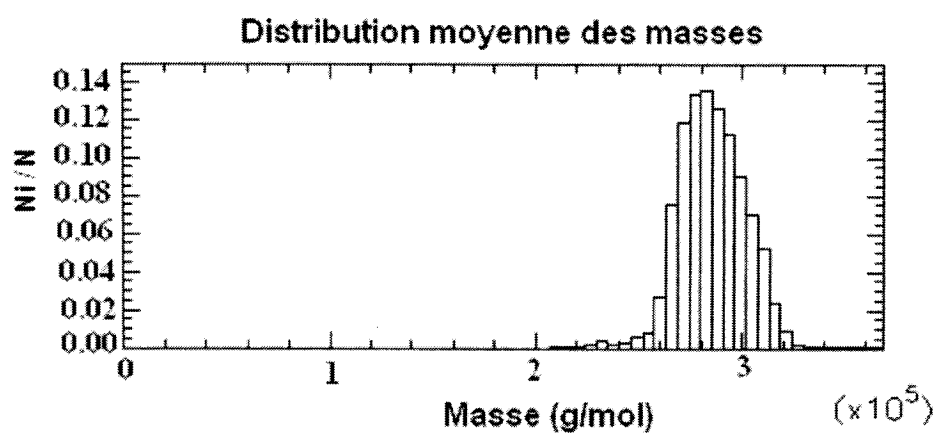


Figure 4.17.a : Estimation de la distribution massique lors de la synthèse des polyactides co-initiée par le pentaérythritol ($k_i = 2 \text{ e-}2$)

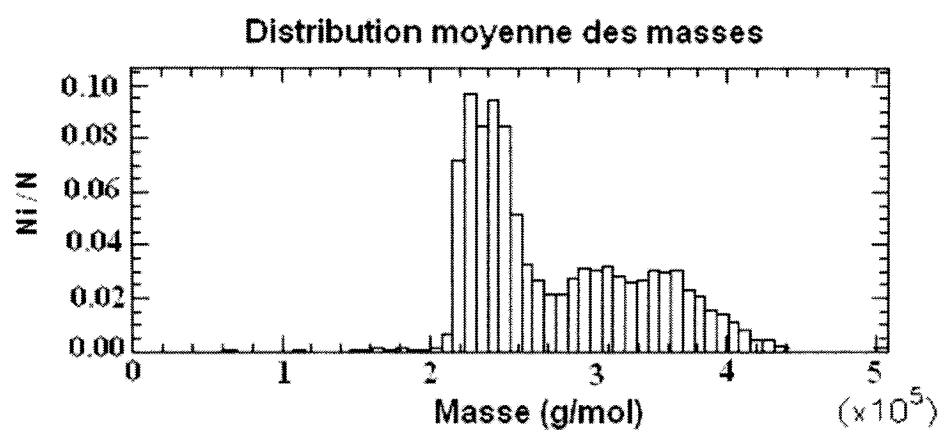


Figure 4.17.b : Estimation de la distribution massique lors de la synthèse des polyactides co-initiée par le pentaérythritol ($k_i = 3 \text{ e-}3$)

CHAPITRE V : DISCUSSION GÉNÉRALE

Dans ce chapitre, nous allons présenter une synthèse des travaux que nous avons effectués sur la modélisation et la simulation stochastique des réactions de polymérisation par ouverture de cycle, et plus précisément sur la polymérisation des lactides. Nous verrons par la suite les différentes constatations que nous avons pu déduire concernant le modèle que nous avons adopté, nous passerons en revue les limites de celui-ci avant de conclure par une brève citation des perspectives de développement de notre travail, une question qui sera abordée avec plus de détails dans le chapitre suivant.

5.1 Synthèse des travaux

Dans ce mémoire nous avons présenté les fondements de l'approche stochastique pour modéliser un système réactionnel chimique, mis en œuvre une application informatique pour simuler l'évolution des réactions de polymérisation par ouverture de cycle ou estimer les constantes de vitesses des réactions, et utilisé cette application au cas de la synthèse des polylactides co-initiée par le biais d'alcools à différentes fonctionnalités.

Il a été choisi de décrire l'évolution d'un système chimique comme une suite d'événements s'y produisant et mettant en jeu les molécules le composant. Le principe fondamental de ce travail est de procéder à une sélection stochastique des événements importants susceptibles de se produire dans un système et de mémoriser l'information sur les changements qui en découlent. Par la suite, en examinant les informations microscopiques recueillies lors des simulations répétées, nous pouvons estimer les valeurs des paramètres macroscopiques du système.

Dans ce mémoire nous ne sommes pas arrêtés seulement au stade de la simulation stochastique; mais nous avons aussi cherché à estimer les constantes

stochastiques régissant l'occurrence des événements, moyennant une comparaison entre les valeurs observées et estimées de paramètres macroscopiques mesurables en laboratoire tel la conversion et la masse moléculaire. Une fois cette estimation faite, nous avons été en mesure de calculer les constantes de vitesse des réactions associées aux étapes principales de la réaction de polymérisation considérée. Notons aussi que lors de cette démarche, il a été question de résoudre les problèmes survenus lors du traitement des informations recueillies pendant l'évolution des systèmes réactionnels chimiques, tel que le codage et la comparaison des chaînes moléculaires apparaissant dans un système chimique, sans oublier le problème de minimisation d'une fonction à plusieurs variables. Le cas concret que nous avons traité est l'estimation des constantes de vitesse des réactions principales se produisant lors de la synthèse des polylactides, co-initiée par différents alcools $R-(OH)_x$ en présence d'octoate d'étain.

5.2 Discussion des résultats

Nous avons utilisé notre code pour estimer les constantes de vitesse des réactions d'initiation, de propagation et de terminaison se produisant lors de la synthèse des polylactides co-initiée par l'alcool benzylique, le 1,4-butanédiol et le pentaérythritol en présence d'octoate d'étain. Les données expérimentales ayant servi dans ces estimations ont été celles de l'article de Korhonen et al. (2001). Plusieurs constatations ont alors été déduites des résultats obtenus.

En premier lieu, il s'est avéré que de meilleures estimations et de bons niveaux de concordance entre les données estimées et expérimentales sont obtenus avec l'alcool benzylique, alcool monofonctionnel, qu'avec le 1,4-butanédiol, alcool bifonctionnel, et le pentaérythritol, alcool à quatre groupements fonctionnels. Cette constatation pourra être expliquée par :

- Le nombre de points d'observations ayant servi dans les estimations pour le cas de l'alcool benzylique est plus grand que celui ayant été utilisé dans les autres estimations. Ceci est dû au fait que les observations ont été réalisées à des intervalles de temps égaux sans égard à la rapidité des réactions de

polymérisation co-initiées par les alcools multifonctionnels. Ce facteur a pour conséquence de réduire le nombre d'observations faites avant la fin de la réaction. Ce constat repose sur le fait que plus grand est le nombre des observations disponibles plus fiable est l'estimation;

- Le fait que l'alcool benzylique est monofonctionnel et que les autres co-initiateurs sont multifonctionnels laisse penser à la véracité de l'hypothèse de l'équiréactivité des sites réactionnels adoptée par notre modèle ainsi que dans la littérature sur les polymères, mais de plus en plus mis en doute notamment par les résultats expérimentaux. Mentionnons aussi le fait que nous avons négligé l'effet de la longueur des chaînes sur les constantes cinétiques des réactions de propagation et de terminaison, et qui reste une hypothèse discutable;
- Le fait que nous avons simplifié notre modèle en éliminant les réactions intramoléculaires et intermoléculaires, notamment celles de la dépolymérisation et de la transestérification, a fait que notre modèle estime d'une manière moins fiable le cas d'un co-initiateur multifonctionnel.

Par ailleurs, un examen minutieux des courbes obtenues par notre modèle pour différentes valeurs des paramètres, démontre qu'elles semblent garder une certaine allure qui ne présente pas les changements brusques observés dans les courbes des données expérimentales : les courbes de notre modèle ne présentent pas des points d'inflexion. Cet effet est associé au temps de mélange : notre modèle retient l'hypothèse du mélange parfait ce qui n'est pas le cas dans la réalité. Pour améliorer le niveau de concordance des courbes expérimentales et estimées nous pouvons translater les données expérimentales dans la direction négative de l'axe du temps ou bien modifier le modèle stochastique pour tenir compte de ce fait.

En outre, nous avons observé que nos courbes estimées réalisent un faible niveau de concordance avec celles issues des résultats expérimentaux au début et à la

fin de la réaction de synthèse, ce que nous pouvons appeler les phases non stationnaires de l'évolution de notre système. Ceci peut être expliqué par l'occurrence de divers événements, comme les réactions secondaires en nombre important, et les phénomènes de dégradation, qui apparaissent à la fin des réactions de polymérisation.

Mentionnons aussi que l'ordre de grandeur pour les constantes de vitesse de réaction estimées reste identique pour les trois co-initiateurs; il est de l'ordre d'une unité pour la propagation, de 10^{-3} pour l'initiation et de 10^{-5} pour la terminaison. Ceci montre que pour une même fonctionnalité, ici la fonction hydroxyle, les constantes de vitesse de réaction sont proches mais restent influencés par le reste de la molécule constituant la chaîne.

Finalement, nous pouvons constater que dans tous les cas étudiés, l'initiation est rapide et est prédominante dans les premières minutes, avant de céder la place à la propagation après, avec une légère participation de la terminaison à la fin de la réaction de synthèse ce qui est en concordance avec la littérature, notamment l'article de Kricheldorf et al. (2000).

Avant de clore cette section, rappelons qu'une validation de l'approche stochastique reste une tâche difficile en soi vu qu'elle doit amener des réponses à des questions du genre : à l'instant t , combien avons-nous de molécules de l'espèce A? La validation à travers des valeurs macroscopiques présente la meilleure solution pour le moment, et pour la mettre en œuvre, il faut mener ses propres mesures dans le laboratoire.

5.3 Limitations des travaux

Une des limitations concerne les réactions principales prises en compte lors de cette étude : nous avons considéré uniquement les réactions d'initiation, de propagation et de terminaison pour modéliser la synthèse des réactions de polymérisation par ouverture de cycle. Un système réactionnel chimique est le théâtre de réactions chimiques et de transformations complexes dont la prise en considération permettra d'améliorer la performance de notre modèle.

Les principales autres limitations de nos travaux sont liées aux capacités de calculs et de mémoire des ordinateurs. En effet, pour traiter un système de 1000 chaînes il faut parfois 12 heures de calcul et en outre, la mémoire des ordinateurs ne peut contenir toutes les informations d'un système réactionnel chimique et plus précisément celles concernant les chaînes formées lors de la polymérisation. Un échantillonnage s'impose et de par cela nous avons une influence sur la précision des résultats. D'autre part, les calculs effectués peuvent induire des dépassements de capacité, les calculs des premiers moments de la distribution massique font intervenir des nombres de plus en plus grands ce qui peut mener à des résultats erronés. Des vérifications entre les résultats obtenus et ceux existants dans la littérature sont donc recommandées.

De plus, le manque de données expérimentales provenant de différentes sources sur la synthèse des polylactides et la contrainte des temps d'exécution nous a mené à restreindre les tests de notre application. Cependant il faut rappeler que les interventions et les changements effectués sur un code peuvent durer le long de sa durée de vie.

5.4 Conclusion

Même si les résultats que nous avons obtenus sont satisfaisants dans l'ensemble, il est possible d'apporter des améliorations au travail effectué jusqu'à présent. Un des points à prendre en compte est de rendre l'application capable de considérer d'autres événements susceptibles de se produire dans le système. Cet ajout présentera une contrainte quant à la recherche des coefficients stochastiques optimaux, car tout ajout d'événement correspond à l'ajout d'une dimension dans l'espace des solutions optimales. Une solution peut résider dans l'utilisation simultanée de l'application développée et des analyses minutieuses aux laboratoires de façon à restreindre l'espace des solutions optimales.

Signalons à la fin que malgré toutes ses limitations, l'étude reste enrichissante, en effet, elle met en application plusieurs aspects de la connaissance scientifique : de la chimie, de la théorie probabiliste, de la théorie des graphes, de l'optimisation et de l'informatique. Elle peut traiter comme nous l'avons mentionné auparavant d'autres

réactions de polymérisation comme celle des caprolactames et les réactions de polymérisation par addition en générale, ceci demande dans certains cas des ajouts minimes dans le code préétabli. Toutefois, notre application implore des développements futurs en vue de la rendre plus performante et plus attrayante. Nous pouvons la percevoir comme une couche de base et que d'autres couches sont nécessaires, les recommandations du chapitre suivant s'inscrivent dans cette perspective.

CHAPITRE VI : CONCLUSION GÉNÉRALE ET RECOMMANDATIONS

Il sera question dans ce chapitre de présenter la conclusion générale issue de cette étude et les perspectives de développement du travail que nous avons effectué sur la simulation stochastique de la polymérisation par ouverture de cycle.

6.1 Conclusion générale

Par ce travail nous avons pu constater que la simulation stochastique permet non seulement de prédire l'évolution d'un système réactionnel chimique mais de plus d'estimer les paramètres caractérisant les événements majeurs qui s'y produisent. En effet, nous avons montré qu'elle est un outil important quant à l'estimation des constantes de vitesse des réactions principales via l'exemple de la synthèse des polylactides co-initiée par des alcools de fonctionnalités différentes. Il a été aussi question de voir comment reproduire l'évolution microscopique d'un système réactionnel chimique par le biais de la modélisation stochastique que nous avons affinée pour représenter la polymérisation par ouverture de cycle des polylactides. Nous avons élaboré un code de calcul permettant de visualiser l'évolution des paramètres macroscopiques du système. Il s'est avéré que notre modèle représente mieux le système dans sa phase stationnaire que dans ses phases transitoires (i.e. au début et à la fin de la polymérisation), et qu'il s'adapte mieux au cas du co-initiateur monofonctionnel qu'à celui du co-initiateur multifonctionnel. Nous avons associé les informations macroscopiques pouvant être mesurées au laboratoire à celles estimées pour déduire la concordance de nos prédictions via le modèle avec la réalité. Nous avons pu en outre dégager les limites d'application de notre étude afin de proposer des perspectives de développement futures pour la rendre plus performante et plus robuste, une question qui sera abordée plus en détail dans la section suivante.

6.2 Perspectives de développements

Vu l'aspect multidisciplinaire que présente ce mémoire, plusieurs ajouts peuvent être énumérés dans le but d'améliorer notre étude :

- Au niveau du modèle stochastique adopté pour représenter l'évolution du système, il pourra être élargi par l'ajout d'autres événements comme la transtérification et la dépolymérisation. En effet le modèle pourra inclure les réactions intramoléculaires, réactions entre groupements fonctionnels d'une même chaîne, et les réactions intermoléculaires, réactions entre groupements fonctionnels de chaînes différentes. Ceci permettra de considérer les chaînes formées comme des entités dynamiques;
- Au niveau des codes élaborés, des optimisations pourront être introduites notamment lors de la comparaison des chaînes formées tout le long de l'évolution du système réactionnel, d'autres améliorations pourront être faites quant au niveau du traitement de l'information recueillie au long de cette évolution ainsi que sur les calculs de recherche du minimum d'une fonction multivariée;
- Au niveau de l'interaction client-application, des interfaces graphiques pourront être établies pour faciliter les actions et les choix de l'utilisateur. En effet l'introduction des données de départ et l'affichage des graphiques pourront nettement être améliorés, un affichage en temps réel pourra être envisagé, ceci dans le but de rendre l'application plus attrayante;
- Au niveau de l'élargissement de l'approche nous pouvons non seulement se limiter au seul volume réactionnel théâtre d'un seul type de polymérisation comme système à modéliser mais introduire le cas de plusieurs types de

monomères et d'initiateurs différents ce qui constituera un système à modéliser plus large;

- Au niveau des résultats de l'application, il sera intéressant d'ajouter des modules traitant de la rhéologie des polymères formés, des modules qui utiliseront l'information à l'échelle microscopique que rend disponible l'application pour construire des estimations macroscopique de la rhéologie du système. Selon cette voie, d'autres sources d'informations pourront être utilisées ou révélées selon le mode d'utilisation choisi.

Donc en conclusion, nous pouvons dire qu'un code de base pour la simulation et l'estimation des paramètres d'un système réactionnel chimique a été réalisé et que les fondements d'une approche pour traiter et élucider les réactions de polymérisation par ouverture de cycle ont été traités. Il reste cependant des améliorations à effectuer pour le rendre plus générale par le biais d'ajout de nouvelles fonctionnalités et plus souple à manipuler moyennant des interfaces graphiques adéquates.

BIBLIOGRAPHIE

ALBERTSSON A.C. and PALMGREN R. *Cationic polymerization of 1,5-dioxepan-2-one with Lewis acids in bulk and solution*, Journal of Macromolecular Science - Pure and Applied Chemistry, 1996, vol.A33, n° 6, Jun, p. 747

CAROTHERS W.H., DOROUGH G.L. and VAN NATTA F.J. *Studies of polymerization and ring formation. X. the reversible polymerization of six-membered cyclic esters*, Journal of American Chemical society, 1932, vol.54, p. 761-772

CORDELLA L.P., FOGGIA P., SANSONE C., VENTO M. *A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, vol.26, n° 10, p.1367-1372.

DUBOIS P., JACOBS C., JEROME R. and TEYSSIÉ *Macromolecular engineering of polylactones and polylactides. 4. Mechanism and kinetics of lactide homopolymerization by aluminum isopropoxide*, Macromolecules, 1991, vol.24, p.2266-2270

DUBOIS C., AIT-KADI A. and TANGUY P.A. *Chemorheology of polyurethane systems as predicted from Monte carlo simulations of their evolutive molecular weight distribution*, Journal of Rheologie, 1998, vol.42(3), p. 435-452.

DUDA A. and PENCZEK S. *Thermodynamics, kinetics, and mechanisms of cyclic esters polymerization*, ACS Symposium Series, 2000, vol. 764, p.160-198

FLORY P.J. *Molecular size distribution in linear condensation polymers*, Journal of American Chemical society, 1936, vol.58, p.1877-1885

GIBSON M.A. and BRUCK J. *Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels*, Journal of Chemical Physics, 2000, vol.A104, p.1876-1889.

GILDING D.K. and REED A.M. *Biodegradable polymers for use in surgery polyglycolic/poly(lactic acid) homo- and copolymers: 1*, Polymer, 1979, vol.20, p.1459-1464

GILLESPIE D.T. *Exact stochastic simulation of coupled chemical reactions*, Journal of Chemical Physics, 1977, vol.81, no25, p. 2340-2361

GILLESPIE D.T. *A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions*, J.Comp. Phys., 1976, vol.22, p.403

GILLESPIE, D.T. *Markov process: An introduction for physical scientists* Academic Press. Philadelphia, PA, 1992

GORDON and TEMPLE *The graph-like state of matter and polymer science*, Chemical Application of Graph Theory, 1975, p. 300-330

JOHNS D.B., LENZ R.W. and LUECKE *Ring opening polymerization*, Elsevier, London, 1984, p.1:464,

KORHONEN H., HELMINEN A. and V.Seppala J. *Synthesis of polylactides in the presence of co-initiators with different numbers of hydroxyl groups*, Polymer, 2001, vol.42, p.7541-7549

KOWALSKI A., LIBISZOWSKI J.; DUDA A. and PENCZEK S. *Kinetics and mechanism of polymerization of cyclic esters*, American Chemical Society, Polymer Preprints, Division of Polymer Chemistry, 1998, vol. 39, n° 2, p.74-75

KOWALSKI A., DUDA A. and PENCZEK S. *Mechanism of cyclic ester polymerization initiated with tin(II) octoate. 2. Macromolecules fitted with tin(II) alkoxide species observed directly in MALDI-TOF spectra*, *Macromolecules*, 2000, vol. 33, n°3, p. 689-695

KRICHELDORF H., KREISER-SAUNDERS I. and BOETTCHER C. *Polylactones: 31. Sn(II)octoate-initiated polymerization of L-lactide: a mechanistic study*, *Polymer*, 1995, vol.36, p.1253-1259

KRICHELDORF H., KREISER-SAUNDERS I. and STRICKER A. *Polylactones 48. SnOct₂-initiated polymerizations of lactide: a mechanistic study*, *Macromolecules*, 2000, vol.33, p.702-709

MACOSKO C.W. and MILLER D.R., *A new derivation of average molecular weights of non linear polymers*, *Macromolecules*, 1976, vol.9, n° 2, p199-206

MURUHAN R., PETZOLD L. R., CAO Y. and GILLESPIE D.T., *Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method*, *Journal of Chemical Physics*, 2003, vol. 119, n° 24, p. 12784-12794

NIJENHUIS A. J., GRIJPMAN D.W and PENNINGA A.J. *Lewis acid catalyzed polymerization of L-lactide. Kinetics and mechanism of the bulk polymerization*, *Macromolecules*, 1992, vol.25, p.6419-6424.

NIJENHUIS A. J., VAN AERT, HUUB A.M., BASTIAANSEN, CEES Du, YING Jun; LEMSTRA and PIET J. *ABA type copolymers of lactide with poly(ethylene glycol). Kinetic, mechanistic, and model studies*, *Macromolecules*, 1995, vol. 28, n° 7, p. 2124

OKADA, M. *Chemical syntheses of biodegradable polymers*, *Progress in Polymer Science (Oxford)*, 2002, vol. 27, n° 1, p. 87-133

PANDIT S., JUVEKAR V.A and TRIVEDI M.K., *Stochastic simulation of polymer reactions*, Chemical Engineering Science, 1993, vol.48, n°7, p.1237-1250.

PENCZEK S. *Cationic ring-opening polymerization (CROP) major mechanistic phenomena*, Journal of Polymer Science, Polymer Chemistry, 2000, vol.38, n°11, p.1919-1933.

PENCZEK S., DUDA A. and KOWALSKI A. *Polymerization of L,L-lactide initiated by aluminum isopropoxide trimer or tetramer*, Macromolecules, 1998, vol. 31, n.°7, p. 2114-2122.

QUIRK R. and LEE B. *Experimental criteria for living polymerizations*, Polymer International, 1992, vol. 27, n° 4, p. 359-367

ROUVRAY D.H. and BALABAN A.T. *Chemical Applications of Graph Theory*, Applications of Graph Theory, Academic Press, New York, 1979, p.177-221

SOSNOWSKI S., GADZINOWSKI M. and SLOMKOWSKI S. *Poly(L,L-lactide) microspheres by ring-opening polymerization*, Macromolecules, 1996, vol. 29, n°13, p. 4556-4564

STOCKMAYER, W.H. *Theory of molecular size distribution and gel formation in branched- chain polymers*, Journal of Chemical Physics, 1943, vol.11,n° 2, p.45-55

STRIDSBERG K. and ALBERTSSON A-C. *Controlled ring-opening polymerization of L-lactide and 1,5-dioxepan-2-one forming a triblock copolymer*, Journal of Polymer Science, Part A: Polymer Chemistry, 2000, vol. 38, n° 10, p.1774-1784

UHRICH K.E, CANNIZZARO S. M., LANGER R. S. and SHAKESHEFF K. M. *Polymeric systems for controlled drug release*, Chemical Reviews, 1999, vol. 99, n°11, p.3181-3198

ULLMANN J.R. *An Algorithm for Subgraph Isomorphism*, Journal of the Association for Computing Machinery, 1976, vol. 23, p. 31-42.

VAN KAMPEN N.G, *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam, 1992

ZHANG J., GAN Z., ZHONG Z. And JING X. *A novel rare earth coordination catalyst for polymerization of biodegradable aliphatic lactones and lactides*, Polymer International, 1998, vol.45, p.60-66

PRESSMAN R.S. *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York, 1997

ANNEXE I

FICHIER ENTREES.TXT

action (0: prevision 1:simulation)

0

valeur de ci

5.03e-8

valeur de cp

1.19e-5

valeur de ct

1.1e-5

valeur du pas dans la direction ci

1e-7

valeur du pas dans la direction cp

1e-4

valeur du pas dans la direction ct

1e-5

nombre de monomeres

100000

nombre co-initiateurs

50

type alcool (1:mono,2:bi,3:quatre) fonctionnel

3

critere de convergence(1: nombre de simulations 2:convergence des moments)

1

nombre de simulations

10

convergence des moments

1e-4

ANNEXE II

EXEMPLE DU FICHIER « OPTIMISATION.TXT »

on debute par le calcul de f aux points du tetrahedre

appel de la simulation avec ci, cp, ct 5.030000E-08 1.190000E-05 1.100000E-05
la valeur de f objective est: 4.614593E-02

appel de la simulation avec ci, cp, ct 1.503000E-07 1.190000E-05 1.100000E-05
la valeur de f objective est: 0.660290

appel de la simulation avec ci, cp, ct 5.030000E-08 1.119000E-04 1.100000E-05
la valeur de f objective est: 3.17320

appel de la simulation avec ci, cp, ct 5.030000E-08 1.190000E-05 2.100000E-05
la valeur de f objective est: 4.788350E-02

optimisation initiee

appel de la simulation avec ci, cp, ct 8.571666E-08 5.649998E-06 1.454167E-05
la valeur de f objective est: 0.196235

appel de la simulation avec ci, cp, ct 1.800834E-08 8.774998E-06 1.777083E-05
la valeur de f objective est: 0.244453

appel de la simulation avec ci, cp, ct 4.005695E-08 9.295832E-06 1.664236E-05
la valeur de f objective est: 8.626959E-02

appel de la simulation avec ci, cp, ct 8.054645E-09 1.641389E-05 1.788658E-05
la valeur de f objective est: 0.190616

appel de la simulation avec ci, cp, ct 2.747015E-08 1.372292E-05 1.705035E-05
la valeur de f objective est: 4.736463E-02

appel de la simulation avec ci, cp, ct 4.532315E-08 1.571945E-05 1.605788E-05
la valeur de f objective est: 0.628947

appel de la simulation avec ci, cp, ct 4.137350E-08 1.090174E-05 1.649624E-05
la valeur de f objective est: 5.598149E-02

le point 1 du prisme de convergence 5.030000E-08 1.190000E-05 1.100000E-05

le point 2 du prisme de convergence 4.137350E-08 1.090174E-05 1.649624E-05

le point 3 du prisme de convergence 2.747015E-08 1.372292E-05 1.705035E-05

le point 4 du prisme de convergence 5.030000E-08 1.190000E-05 2.100000E-05

le point de convergence p 5.030000E-08 1.190000E-05 1.100000E-05

iterations faites 7

appel de la simulation avec ci, cp, ct 5.030000E-08 1.190000E-05 1.100000E-05
la valeur de f objective est: 4.705641E-02

ANNEXE III

EXTRAIT DU FICHIER « POLYFINAL.TXT »

action (0: prevision 1:simulation) :

1.00000

valeur de ci :

5.030000E-08

valeur de cp :

1.190000E-05

valeur de ct :

1.100000E-05

nombre de monomeres :

100000.

nombre co-initiateurs :

50.0000

type alcool (1:mono,2:bi,3:thetra) fonctionnel :

3.00000

composition finale:

numero de la simulation: 1

nombre d especes différents: 50

masse (g/mol)	fréquence
---------------	-----------

229484.0	1
----------	---

230780.0	1
----------	---

232364.0	1
----------	---

233228.0	1
----------	---

234956.0	1
----------	---

238412.0	1
----------	---

239564.0	1
----------	---

242300.0	1
----------	---

244316.0	1
----------	---

247484.0	1
----------	---

250652.0	1
----------	---

250652.0	1
----------	---

252092.0	1
----------	---

254828.0	1
----------	---

256844.0	1
----------	---

259292.0	1
----------	---

259292.0	1
----------	---

259436.0	1
----------	---

261740.0	1
----------	---

262460.0	1
----------	---

262604.0	1
----------	---

266780.0	1
266780.0	1
267068.0	1
268364.0	1
272396.0	1
280028.0	1
284492.0	1
285644.0	1
286940.0	1
292268.0	1
295148.0	1
296156.0	1
310700.0	1
317036.0	1
319196.0	1
320492.0	1
321212.0	1
324092.0	1
331868.0	1
334748.0	1
334892.0	1
340796.0	1
343388.0	1
344252.0	1
344972.0	1
347132.0	1
352748.0	1
362540.0	1
374492.0	1

numero de la simulation: 2
nombre d especes différents: 50
masse (g/mol) fréquence

226604.0	1
232796.0	1
235820.0	1
239420.0	1
245468.0	1
245756.0	1
246620.0	1

.....etc.

ANNEXE IV

EXTRAIT DU FICHIER « PROBA.TXT »

action (0 : prevision 1 :simulation) :

1.00000

valeur de ci :

5.030000^E-08

valeur de cp :

1.190000^E-05

valeur de ct :

1.100000^E-05

nombre de monomeres :

100000.

Nombre co-initiateurs :

50.0000

type alcool (1 :mono,2 :bi,3 :tetra) fonctionnel :

3.00000

probabilités des événements

numero de la simulation : 1

temps(s)	probabilite1	probabilite2	probabilite3	probabilite4
0.0	1.000000	0.000000	0.000000	0.000000
68.5	0.004584	0.371238	0.000004	0.624174
97.0	0.003574	0.435641	0.000004	0.560780
119.8	0.002820	0.483076	0.000005	0.514100
140.1	0.002171	0.523977	0.000005	0.473846
156.9	0.001897	0.537605	0.000006	0.460493
174.7	0.001704	0.545420	0.000006	0.452870
190.9	0.001551	0.550345	0.000006	0.448098
206.7	0.001333	0.560577	0.000006	0.438084
222.3	0.001271	0.558501	0.000006	0.440222
238.5	0.001188	0.558088	0.000006	0.440717
254.7	0.001109	0.557395	0.000007	0.441490
270.8	0.001032	0.556417	0.000007	0.442544
287.3	0.000999	0.551628	0.000007	0.447366

303.2	0.000947	0.548395	0.000007	0.450652
319.5	0.000857	0.548352	0.000007	0.450784
336.1	0.000809	0.544575	0.000007	0.454609
354.6	0.000780	0.538862	0.000008	0.460351
372.1	0.000716	0.536211	0.000008	0.463066
389.9	0.000671	0.531597	0.000008	0.467724
407.6	0.000612	0.528275	0.000008	0.471105
424.8	0.000539	0.526143	0.000008	0.473310
442.9	0.000515	0.518957	0.000009	0.480520
461.6	0.000491	0.511436	0.000009	0.488065
480.4	0.000467	0.503556	0.000009	0.495968
499.5	0.000416	0.498320	0.000009	0.501254
520.0	0.000367	0.492603	0.000009	0.507020
540.2	0.000347	0.483460	0.000010	0.516184
562.4	0.000301	0.476742	0.000010	0.522947
585.3	0.000282	0.466603	0.000010	0.533105
608.7	0.000228	0.460131	0.000011	0.539630
633.3	0.000200	0.450198	0.000011	0.549592
658.0	0.000183	0.438237	0.000011	0.561568
683.3	0.000167	0.425586	0.000012	0.574235
711.0	0.000161	0.410863	0.000012	0.588963
740.8	0.000146	0.396672	0.000012	0.603170
771.6	0.000122	0.382874	0.000013	0.616992
806.3	0.000116	0.365557	0.000013	0.634314
842.3	0.000110	0.347240	0.000013	0.652636
882.4	0.000104	0.327833	0.000014	0.672049
926.2	0.000091	0.308381	0.000014	0.691514
974.3	0.000084	0.286433	0.000015	0.713468
1025.8	0.000071	0.264083	0.000015	0.735830

1085.6	0.000059	0.240012	0.000016	0.759913
1157.5	0.000052	0.212242	0.000016	0.787690
1237.8	0.000045	0.183480	0.000017	0.816458
1331.6	0.000038	0.152538	0.000018	0.847407
1459.2	0.000029	0.119157	0.000018	0.880795
1635.6	0.000021	0.082633	0.000019	0.917328
1948.9	0.000011	0.043160	0.000019	0.956809

numero de la simulation : 2

temps(s)	probabilite1	probabilite2	probabilite3	probabilite4
0.0	1.000000	0.000000	0.000000	0.000000
66.0	0.004458	0.380284	0.000004	0.615254
93.5	0.003574	0.435641	0.000004	0.560780
118.8	0.003046	0.464543	0.000005	0.532406
141.4	0.002461	0.500571	0.000005	0.496963
161.2	0.002328	0.503162	0.000005	0.494505

.....etc.

ANNEXE V

EXTRAIT DU FICHIER « MOMENTS.TXT »

action (0: prevision 1:simulation) :

1.00000

valeur de ci :

5.030000E-08

valeur de cp :

1.190000E-05

valeur de ct :

1.100000E-05

nombre de monomeres :

100000.

nombre co-initiateurs :

50.0000

type alcool (1:mono,2:bi,3:thetra) fonctionnel :

3.00000

évolution des moments

numero de la simulation: 1

temps(s)	moment1	moment2	moment3
0.0	0.00	0.00	0.00
68.5	9390.91	12198.80	13722.11
97.0	15818.32	18959.07	21655.01
119.8	20750.98	25292.04	28797.30
140.1	25698.96	30929.31	35264.26
156.9	30653.00	36098.16	40504.90
174.7	36650.00	41573.37	46348.04
190.9	42647.00	47336.30	52114.78
206.7	47664.90	53151.94	58224.30
222.3	53539.51	58919.98	64099.21
238.5	59414.12	64852.12	70155.19
254.7	65288.73	70730.20	76003.87
270.8	71163.34	76876.59	82313.59

287.3	77037.96	83010.50	88624.72
303.2	82912.57	89256.07	95237.72
319.5	87024.80	95344.26	101561.01
336.1	92781.92	101252.81	107730.50
354.6	98539.04	106998.80	113744.92
372.1	104296.20	112760.82	119728.64
389.9	110053.30	118588.60	125833.01
407.6	115810.40	124284.10	131709.73
424.8	121567.52	129959.40	137500.70
442.9	127324.64	135615.54	143240.50
461.6	133081.80	141192.70	148785.70
480.4	138838.90	147047.80	154859.80
499.5	144596.00	152748.20	160583.30
520.0	150353.12	158324.00	166057.14
540.2	156110.23	163981.60	171691.34
562.4	161867.40	169606.10	177242.40
585.3	167624.50	175336.24	182970.00
608.7	173381.60	181098.62	188777.60
633.3	179138.71	186733.71	194337.50
658.0	184895.84	192365.10	199869.50
683.3	190653.00	198003.80	205408.80
711.0	196410.10	203579.90	210827.73
740.8	202167.20	209147.40	216213.71
771.6	207924.31	214787.10	221744.04
806.3	213681.43	220429.80	227266.93
842.3	219438.60	226096.04	232847.80
882.4	225195.70	231795.01	238496.40
926.2	230952.80	237451.70	244053.10
974.3	236709.92	243132.50	249657.20

1025.8	242467.04	248883.42	255395.04
1085.6	248224.20	254528.10	260923.60
1157.5	253981.30	260183.00	266487.34
1237.8	259738.40	265795.40	271967.81
1331.6	265495.53	271508.20	277643.71
1459.2	271252.62	277197.21	283269.80
1635.6	277009.74	282930.40	288987.43
1948.9	285788.00	291644.53	297629.81

numero de la simulation: 2

temps(s)	moment1	moment2	moment3
0.0	0.00	0.00	0.00
66.0	9134.35	10687.61	11888.56
93.5	15060.80	17897.32	20358.77
118.8	20750.98	24627.68	28289.42
141.4	26255.20	30822.20	35706.48

.....etc.

ANNEXE VI

CODE DU PROGRAMME

```

! -----
!
! fichier control.f95
!         contient la procédure de control de l'évolution du système
!
! Created on Tue Jun 14 21:41:04 2005
! Copyright (c) 2005 foutlane. All rights reserved.
!
! -----
Module controle
use type_de_donnees
contains
    subroutine
subcontrol(compteur,nombre_monomeres,nombre_chaines,conversion,n&
ombre_chaines_différentes,convers,prob1,prob2,prob3,prob4,p1,p2,p3,p4,Mo,tt,temps&
,tab_chaines,MMn,MMw,MMz,freq1,freq2,freq3)
    type(chaine),Dimension(:),pointer::tab_chaines
    integer::
compteur,nombre_monomeres,nombre_chaines,nombre_chaines_différentes&
,Mo,step1,kk,nombre,freq1,freq2,freq3
    real ::conversion,p1,p2,p3,p4,temps
    real,Dimension(:)::prob1,prob2,prob3,prob4,tt,MMw,MMn,MMz
    real,Dimension(:)::convers
    real ::step,ss,ss2,ss3,w
        ss=0
        ss2=0
        ss3=0
        nombre=0
        step=0.02
        step1=int(Mo*step)
        if(modulo(Mo-nombre_monomeres,step1)==0.and.conversion>0) then
            compteur=compteur+1
            convers(compteur)=conversion
            prob1(compteur)=p1
            prob2(compteur)=p2
            prob3(compteur)=p3
            prob4(compteur)=p4
            tt(compteur)=temps
        do kk=1,nombre_chaines_différentes

```

```

        w=Masse_Molaire(tab_chaines(kk),Masse_Monomere)
        ss=ss+ w*tab_chaines(kk)%frequence
        ss2=ss2 + w*w*tab_chaines(kk)%frequence
        ss3=ss3 + (w**3)*tab_chaines(kk)%frequence
        nombre=nombre+tab_chaines(kk)%frequence
    end do

    MMn(compteur)=ss/nombre
    MMw(compteur)=ss2/ss
    MMz(compteur)=ss3/ss2
end if

end subroutine subcontrol

end module controle

! -----
! fichier statistiques.f95
!   contenant les procédures de calcul des espérances des paramètres
!   de l'évolution du système:
!       probabilités des événements
!       premiers moments de la distribution massique
!       Distribution massique
!   Created on Tue Jun 14 22:36:23 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module traitements
use type_de_donnees
contains

    integer function nombre_unites(tab_chaines,nombre)
        type(chaine),Dimension(:),Pointer::tab_chaines
        integer kk,h,nombre
        kk=0
        do h=1,nombre
            kk=kk+tab_chaines(h)%frequence

```

```

end do
nombre_unites=kk
end function nombre_unites

subroutine subCalculMoyen (replications, tempstotal, converstotal, Mwttotal,
tempsmoyen,convers,Moment,nbpoints,prob1total,prob2total,prob3total,prob4total,prob
1moyen,prob2moyen,prob3moyen,prob4moyen)
integer h,j,k,replications,nbpoints
real,Dimension(:,:):tempstotal,converstotal,Mwttotal
real,Dimension(:,:):prob1total,prob2total,prob3total,prob4total
real,Dimension(100):tempsmoyen,convers,Moment
real,Dimension(100): prob1moyen, prob2moyen, prob3moyen, prob4moyen
h=100
do k=1,replications
do j=1,100
if(converstotal(k,j)==-1) then
if(j-1<h) then
h=j-1
end if
exit
end if
end do
end do
nbpoints=h
do k=1,h
tempsmoyen(k)=sum(tempstotal(1:replications,k))/replications
convers(k)=sum(converstotal(1:replications,k))/replications
Moment(k)=sum(Mwttotal(1:replications,k))/replications
prob1moyen(k)=sum(prob1total(1:replications,k))/replications

```

```

        prob2moyen(k)=sum(prob2total(1:replications,k))/replications
        prob3moyen(k)=sum(prob3total(1:replications,k))/replications
        prob4moyen(k)=sum(prob4total(1:replications,k))/replications
    end do
end subroutine subCalculMoyen

subroutine calcul_distribution(nombre,tableau,tabC2,elements_differents,maxi)
    integer nombre,kk,jj,h
    integer,Dimension(:)::elements_differents
    type(pointurchaine),Dimension(:),pointer::tabC2
    real maxi
    real,Dimension(:)::tableau
    tableau=0
    do kk=1,nombre
        do jj=1,elements_differents(kk)
            do h=59,0,-1
                if(Masse_Molaire(tabC2(kk)%tabchaines(jj),Masse_Monomere)>(h*maxi/60))
then
                    tableau(h+1)=tableau(h+1)+tabC2(kk)%tabchaines(jj)%frequence
                    exit
                end if
            end do
        end do
    end do
    tableau(:)=(int(tableau(:)/nombre))*1.0
end subroutine calcul_distribution
end Module traitements

```

```

! -----
! fichier Simulation.f95
!   contenant les procédures de réalisation de:
!       - une simulation
!       - n simulations afin de satisfaire le critère d'arrêt
!   Created on Tue Jun 14 09:07:16 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module simulation
    use operations
    use controle
    use resultat_simulation
    use traitements
    integer critere
    integer MaxSimulations
    real precisio
    contains
    subroutine simuler(
rates,chai,Io,Mo,convers,prob1,prob2,prob3,prob4,tt,MMn,MMw,MMz&
,nombre_chaines_différentes,compteur)
        real rates(3)
        type(chaine):: chain,chai
        integer ::h=0,k=0,j=0,kk,nombre_chaines_différentes,replications=0
        integer liai,values(8),new_seed(2)
        character*8 date,time,zone
        integer:: compteur,Io,Mo,indices(3)=0
        integer:: freq1,freq2,freq3,nombre=0,nombre2=0,sortie
        real convers(100),prob1(100),prob2(100),prob3(100),prob4(100),tt(100)&
,MMn(100),MMw(100),MMz(100)

```

```

real p1,p2,p3,pt,conversion,temps
real u,uk,uj,dt,aw,pi
real ::ss=0,ss2=0,ss3=0,w=0
integer,Dimension(:),Pointer::tab_sites,tab_sitesNonI
integer nombre_sitesI,nombre_sitesA,nombre_sitesNonI
integer diffe
convers=-1
conversion=0.
tempsmoyen=0
nombre_monomeres=Mo
nombre_chaines=0
nombre_chaines_différentes=0
liai=chai%nombre_liaison
call date_and_time(date,time,zone,values)
new_seed(1)=values(8)+1000*(values(7)+60*(values(6)+60*values(5)))
call random_seed(PUT=new_seed(1:1))
allocate(tab_sitesNonI(Io))
allocate(tab_sites(Io))
compteur=1
sortie=0
freq1=0
freq2=0
freq3=0
allocate(tab_chaines(Io))
nombre_monomeres=Mo
nombre_chaines=0
nombre_chaines_différentes=0
convers(1)=0.
dt=0.01

```

```

temps=0
nombre_sitesA=0
nombre_sitesI=0
nombre_sitesNonI=0
prob1(1)=dble(Io*liai-nombre_sitesI)*dble(nombre_monomeres)
prob2(1)=0
prob3(1)=0
prob4(1)=0
MMn=0
MMw=0
MMz=0
prob1(1)=1-prob4(1)
tt(1)=0
conversion=-1
tab_sitesNonI=0
do while(conversion<0.99)
    conversion=(Mo-nombre_monomeres)/(Mo*1.0)
    p1=rates(1)*dble(Io*liai-nombre_sitesI)*dble(nombre_monomeres)
    p2=rates(2)*dble(nombre_sitesA)*dble(nombre_monomeres)
    p3=rates(3)*dble(nombre_sitesA)
    pt=p1+p2+p3+1/dt
    if(p1<0 .or. p2<0 .or. p3<0) then
        pause
        print*,"valeurs negatifs"
    print*,nombre_sites,nombre_chaines,nombre_chaines_différentes,p1,p2,p3,p4,fr
eq1,freq2,freq3
    print*,tab_sites(:)
end if
p1=p1/pt

```

```

p2=p2/pt
p3=p3/pt
p4=1/dt/pt
call random_number(u)
if(u<p1) then
    freq1=freq1+1
    if(freq1<=(Io*liai)) then
        chain=chai
        nombre_sitesI=nombre_sitesI+1
        if(freq1==1) then
            call random_number(u)
            j=int(liai*u)+1
            chain%frequence=1
            chain%tab_liaison(j)%etat=1
            call
subinitiation(tab_chaines,nombre_chaines_différentes,nombre_chaines,Io,chain)
        else
            call random_number(aw)
            pi=nombre_sitesNonI*1.0/((Io-
nombre_chaines)*liai+nombre_sitesNonI)
            if(aw>pi) then
                call random_number(u)
                j=int(liai*u)+1
                chain%frequence=1
                chain%tab_liaison(j)%etat=1
call subinitiation(tab_chaines,nombre_chaines_différentes,nombre_chaines,Io,chain)
            else
                call random_number(aw)
                j=int(tab_sitesNonI(nombre_chaines_différentes)*awk)+1

```

```

k=case_reaction(tab_sitesNonI,j,nombre_chaines_différentes)
    if(k==1) then
        limite1=0
    else
        limite1=tab_sitesNonI(k-1)
    end if
    limite2=tab_sitesNonI(k)
    diffe=limite2-limite1
    call subinitiation2( tab_chaines,
nombre_chaines_différentes, k, diffe)
    end if
end if
call
subajustement(tab_sites,tab_chaines,nombre_chaines_différentes)
    call subajustementIni (tab_sitesNonI,
tab_chaines,nombre_chaines_différentes)
    nombre_sitesNonI=tab_sitesNonI(nombre_chaines_différentes)
    end if
else if (u<(p1+p2) )then
    freq2=freq2+1
    if(nombre_chaines>0) then
        call subcontrol (compteur,
nombre_monomeres, nombre_chaines, conversion, nombre_chaines_différentes,
convers,prob1,prob2,prob3,prob4,p1,p2,p3,p4,Mo,tt,temps,tab_chaines,MMn,MMw,M
Mz,freq1,freq2,freq3)

        call random_number(uk)
        call random_number(uj)
        j=int(tab_sites(nombre_chaines_différentes)*uk)+1

```

```

        k=case_reaction(tab_sites,j,nombre_chaines_différentes)
        j=int(liai*uj)+1
        call
subpropagation(tab_chaines,k,j,nombre_chaines_différentes,sortie,nombre_monomeres)
        call
subajustement(tab_sites,tab_chaines,nombre_chaines_différentes)
        call
subajustementIni(tab_sitesNonI,tab_chaines,nombre_chaines_différentes)

        end if
    else if (u<(p1+p2+p3)) then
        freq3=freq3+1
        if(nombre_chaines>0) then
            call random_number(uk)
            call random_number(uj)
            j=int(tab_sites(nombre_chaines_différentes)*uk)+1
            k=case_reaction(tab_sites,j,nombre_chaines_différentes)
            j=int(liai*uj)+1
            call subtransfert ( tab_chaines,k, j,
nombre_chaines_différentes, sortie)
            call subajustement (tab_sites,tab_chaines,
nombre_chaines_différentes)
            call subajustementIn i(tab_sitesNonI,tab_chaines,
nombre_chaines_différentes)
        end if
    else
        temps=temps+dt

```

```

end if
if (sortie==1) then
    compteur=compteur+1
    convers(compteur)=conversion
    prob1(compteur)=p1
    prob2(compteur)=p2
    prob3(compteur)=p3
    prob4(compteur)=p4
    tt(compteur)=temps
    print*,temps
    print*,tt(:)
    exit
end if
if(nombre_chaines_différentes>0) then
    nombre_sitesA=tab_sites(nombre_chaines_différentes)
    if(nombre_sitesA==0) then
        print*,"pas de sites actifs"
        compteur=compteur+1
        convers(compteur)=conversion
        prob1(compteur)=p1
        prob2(compteur)=p2
        prob3(compteur)=p3
        prob4(compteur)=p4
        tt(compteur)=temps
        exit
    end if
    if(nombre_monomeres==0) then
        print*,"pas de monomeres"
        compteur=compteur+1

```

```

                                convers(compteur)=conversion
                                prob1(compteur)=p1
                                prob2(compteur)=p2
                                prob3(compteur)=p3
                                prob4(compteur)=p4
                                tt(compteur)=temps
                                end if
                        end if

end do

do kk=1,nombre_chaines_différentes
    w=Masse_Molaire(tab_chaines(kk),Masse_Monomere)
    ss=ss+ w*tab_chaines(kk)%frequence
    ss2=ss2 + w*w*tab_chaines(kk)%frequence
    ss3=ss3 + (w**3)*tab_chaines(kk)%frequence
    nombre=nombre+tab_chaines(kk)%frequence
end do

MMn(compteur)=ss/nombre
MMw(compteur)=ss2/ss
MMz(compteur)=ss3/ss2
print*, 'Mn',MMn(compteur), 'Mw',MMw(compteur), 'Mz',MMz(compteur), 'tem',t
emps,nombre_monomeres
deallocate(tab_sites)
end subroutine

subroutine nsimuler(chai,Mo,Io,rates)
    real rates(3)
    type(chaine):: chai
    integer::essai=1
    integer compteur

```

```

logical fin
real conver(100),prob1(100),prob2(100),prob3(100),prob4(100),tt(100)
real MMw(100),MMn(100),MMz(100)
integer nombre_chaines_différentes
integer Io,Mo
real :: epsilo,Mnold,Mwold,Mzold
if(essai==1) then
  allocate (Mwtotat(5000,100))
  allocate (Mntotat(5000,100))
  allocate (Mztotat(5000,100))
  allocate(converstat(5000,100))
  allocate(prob1totat(5000,100))
  allocate(prob2totat(5000,100))
  allocate(prob3totat(5000,100))
  allocate(prob4totat(5000,100))
  allocate(tempstat(5000,100))
  allocate(tab_chaines2(5000))
else
  Mwtotat=0
  Mntotat=0
  Mztotat=0
  converstat=0
  prob1totat=0
  prob2totat=0
  prob3totat=0
  prob4totat=0
  tempstat=0
  nullify(tab_chaines2)
end if

```

```

essai=essai+1
maxi=0
replications=0
epsilo=10
Mnold=0
Mwold=0
Mzold=0
fin=.false.
do, while(fin==.false.)
    replications=replications+1
    call simuler
(rates,chai,lo,Mo,conver,prob1,prob2,prob3,prob4,tt,MMn,MMw,MMz,nombre_chaines
_differentes&
,compteur)
    Mntotal(replications,:)=MMn(:)
    Mwttotal(replications,:)=MMw(:)
    Mzttotal(replications,:)=MMz(:)
    prob1total(replications,:)=prob1(:)
    prob2total(replications,:)=prob2(:)
    prob3total(replications,:)=prob3(:)
    prob4total(replications,:)=prob4(:)
    tempsttotal(replications,:)=tt(:)
    conversttotal(replications,:)=conver(:)
    allocate(tab_chaines2(replications)%tabchaines(nombre_chaines_differentes))
    tab_chaines2(replications)%tabchaines(:)=tab_chaines(:)
    elements_differents(replications)=nombre_chaines_differentes
    maxi=max(maxi,Masse_Molaire(tab_chaines(nombre_chaines_differentes),Mass
e_Monomere))
    Mnnew=MMn(compteur)

```

```

Mwnew=MMw(compteur)
Mznew=MMz(compteur)
epsilo=max(abs(Mnnew-Mnold)/Mnold,abs(Mwnew-
Mwold)/Mwold,abs(Mznew-Mzold)/Mzold)
print*,"epsilon",epsilo
Mnold=Mnnew
Mwold=Mwnew
Mzold=Mznew
if(critere==1.and.replications==MaxSimulations) then
    fin=.true.
else if(critere==2.and.epsilo<precisio) then
    fin=.true.
end if
end do
call
subCalculMoyen(replications,tempstotal,converstotal,Mwttotal,tempmoyen&
,convers,Moment,nbpoints_simu,prob1total,prob2total,prob3total,prob4total,prob1moye
n,prob2moyen,prob3moyen,prob4moyen)

call calcul_distribution (replications,tableau, tab_chaines2,elements_differents,
maxi)
end subroutine
end Module Simulation

```

```

! -----
!fichier resultat_simulation.f95
!          contient les déclarations des tableaux de stockage
!          de l'information sur le déroulement des simulations
!   Created on Wed Jun 15 14:26:19 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module resultat_simulation
use type_de_donnees
integer replications
integer,Dimension(5000):: elements_differeents
real
,Dimension(:,,:),allocatable::converstotal,prob1total,prob2total,prob3total,prob4total,tem
pstotal
real, Dimension(:,,:),allocatable ::Mwttotal,Mzttotal,Mnttotal
real tempsmoyen(100),convers(100),Moment(100)
real prob1moyen(100),prob2moyen(100),prob3moyen(100),prob4moyen(100)
type(pointeurchaine),Dimension(:),pointer::tab_chaines2
real maxi
real tableau(60)
integer nbpoints_simu
integer ::convergence_atteinte=1
end Module

```

```

! -----
!fichier types.f95
!   contient les définitions des types utilisés par le code,
!   déclaration des données du système et des fonctions élémentaires:
!       calcul de la masse molaire
!       calcul de masse de noyau
!       conversion chaîne vers matrice
!       vérification de l'égalité de 2 noyaux
!   Created on Tue Jun 14 17:23:44 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----
Module type_de_donnees

```

```

! déclaration de données du système

```

```

integer nombre_chaines

```

```

integer nombre_monomeres

```

```

real ::Masse_Monomere=72

```

```

!définition du type noyau CHNO

```

```

type noyau

```

```

integer*1 ::carbon=0

```

```

integer*1 ::hydrogene=0

```

```

integer*1 ::azote=0

```

```

integer*1 ::oxygene=0

```

```

end type noyau

```

```

!définition du type coordonnes

```

```

type coordonnes

```

```

integer*1 ligne

```

```

integer*1 colonne

```

```

end type coordonnees
!définition type site
type site
integer numero
integer branche
end type site
!définition du type liaison
type liaison
integer*1 indice__coord
integer*1 etat
integer nbre__unite__repetitif
end type liaison
!définition du type chaîne
type chaine
integer*1 type__initialiseur,nombre__liaison
integer frequence
type(liaison),Dimension(20):: tab__liaison
end type chaine
! définition du type pointeur chaine
type pointeurchaine
type(chaine),Dimension(:),pointer::tabchaines
end type pointeurchaine

!définition du type de l'initiateur
type initialiseur
integer ::nombre__noyaux=0
integer ::nombre__liaison__variables=0
integer ::nombre__liaison__fixes=0
type(noyau),Dimension(30)::tab__noyaux

```

```

type(coordonnes),Dimension(20)::tab_coord_fixes,tab_coord_variables
end type initialiseur

```

```

!définition du type pointeur de l'initiateur

```

```

type(initialiseur), Dimension(:),Pointer::tab_initialiseurs

```

```

type(chaine),Dimension(:),Pointer::tab_chaines

```

```

contains

```

```

! constructeur de noyaux

```

```

type(noyau) function constructeur_noyau(C,H,O,N)

```

```

integer C,H,O,N

```

```

constructeur_noyau%carbon=C

```

```

constructeur_noyau%hydrogene=H

```

```

constructeur_noyau%oxygene=O

```

```

constructeur_noyau%azote=N

```

```

end function constructeur_noyau

```

```

!fonction vérifiant l'egalité de 2 noyaux

```

```

logical function StEgales(noy1,noy2)

```

```

type(noyau) noy1,noy2

```

```

StEgales=(noy1%carbon==noy2%carbon).and.(noy1%hydrogene==noy2%hydrogene)&

```

```

.and.(noy1%oxygene==noy2%oxygene).and.(noy1%azote==noy2%azote)

```

```

end function StEgales

```

```

! fonction calcul de masse de noyau

```

```

real function Masse_Noyau(noy)

```

```

type(noyau) noy

```

```

Masse_Noyau=12*noy%carbon+16*noy%oxygene+1*noy%hydrogene+18*noy%azote

```

```

end function Masse_Noyau

```

```

! subroutine convertir chaine vers matrice

```

```

subroutine ChainetoMatrice(chai,n)
type(chaine) chai
integer k,n
type(initialiseur) init
integer,Dimension(:,:),Pointer:: matrice
init= tab_initialiseurs(chai%type_initialiseur)
allocate(matrice(init%nombre_noyaux,init%nombre_noyaux))
matrice=0
do k=1, init%nombre_liaison_fixes
matrice(init%tab_coord_fixes(k)%ligne,init%tab_coord_fixes(k)%colonne)=1
matrice(init%tab_coord_fixes(k)%colonne,init%tab_coord_fixes(k)%ligne)=1
end do
do k=1, init%nombre_liaison_variables
matrice(init%tab_coord_variables(k)%ligne,init%tab_coord_variables(k)%colonne)=&
chai%tab_liaison(k)%nbre_unite_repetitif
matrice(init%tab_coord_variables(k)%colonne,init%tab_coord_variables(k)%ligne)=&
chai%tab_liaison(k)%nbre_unite_repetitif
end do
do k=1, init%nombre_noyaux
write(n,*) matrice(k,:)
end do
deallocate (matrice)
end subroutine ChainetoMatrice

!fonction retournant le site libre de la chaîne
integer function est_libre(chai)f
type(chaine) chai
integer i,f
f=chai%nombre_liaison

```

```

do i=1,f
if(chai%tab_liaison(i)%etat==0) then
est_libre=i
return
end if
end do
est_libre=-1
end function est_libre

```

```

! fonction calculant la masse molaire d'l chaîne
real function Masse_Molaire(chai,Masse_Monomere)
type(chaine) chai
real ::Masse_Monomere
real::m
type(initialiseur) init
integer i
m=0
init=tab_initialiseurs(chai%type_initialiseur)
do i=1,chai%nombre_liaison
m=m+(chai%tab_liaison(i)%nbre_unite_repetitif*Masse_Monomere)
end do
do i=1,init%nombre_noyaux
m=m+Masse_Noyau(init%tab_noyaux(i))
end do
Masse_Molaire=m
end function Masse_Molaire
end Module type_de_donnees

```

```

! -----
!fichier resultat_experimental.f95
!          contient les déclarations des variables
!          pour stocker les résultats expérimentaux
!
!   Created on Thu Jun 16 02:23:54 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module resultat_experimental

real,Dimension(100,10)::tabX
real,Dimension(100)::tabtemps
integer nbre_source
integer nbre_observations
real,Dimension(10)::Ma
end Module

! -----
!
!fichier prevision.f95
!          contient la procédure de prévision de l'évolution
!          du systeme reactionnel
!   Created on Wed Jun 15 18:32:32 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module prevision
    use simulation
    contains
    subroutine prevoir(chai,Mo,Io,rates)

```

```

        type(chaine)::chai
        integer Mo, Io
        real rates(3)
        call nsimuler(chai, Mo, Io, rates)
    end subroutine
end module prevision

! -----
!
!fichier Minimisation.f95
!
!      contient les procédures de calcul de:
!
!      la valeur de la fonction objective en un point x&
!
!      la recherche du minimum d'une fonction
!
!   Created on Wed Jun 15 21:25:22 2005
!
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module Minimisation
    use simulation
    use donneesglobal
    use resultat_experimental
contains
    real function funk2(x)
        implicit none
        real x(3)
        integer j
        write(14,*) "*****"
        write(14,*) " "
        write(*,*) "appel de la simulation avec ci, cp, ct", x(1), " ", x(2), " ", x(3)
        write(14,*) "appel de la simulation avec ci, cp, ct", x(1), " ", x(2), " ", x(3)
        tempsmoyen=0

```

```

    convers=0
    Moment=0
    call nsimuler(chai,Mo,Io,x)
    funk2=objective(Ma)
    write(6,*) "les k correspondants",(x(j),j=1,3),funk2
    write(14,*) "la valeur de f objective est:",funk2
    return
end function funk2

real function objective(Ma)
integer j,kk,h
    real ,Dimension(100,10)::datasim
    real ,Dimension(10)::valobj,Ma
    real ,Dimension(100,10)::datasimu

    print*,nbre_observations,nbre_source
    datasim=0
    valobj=0
    datasimu(:,1)=convers(:)
    datasimu(:,2)=Moment(:)
    j=1
    kk=2
    do, while(j<nbre_observations )
        j=j+1
        do,while(tempismoyen(kk)<tabtemps(j) .and. kk<=nbpoints_simu)

            kk=kk+1
        end do
        if(kk>nbpoints_simu) then

```

```

        exit
    end if
    do h=1,nbre_source
        datasim(j,h)=(datasimu(kk,h)-datasimu(kk-
1,h))/(tempsmoyen(kk)-tempsmoyen(kk-1))*(tabtemps(j)-tempsmoyen(kk-1))+
datasimu(kk-1,h)
    end do
end do
do h=1,nbre_source
    valobj(h)=sum(((datasim(1:nbre_observations,h)-
tabX(1:nbre_observations,h))/&
Ma(h))**2)
end do
print*,"valobj",valobj(1:nbre_source),Ma(1:nbre_source)
objective=(1.0/nbre_source)*sum(valobj(1:nbre_source))
end function

SUBROUTINE amoeba(p,y,mp,np,ndim,ftol,funk,iter,amotry)
INTEGER iter,mp,ndim,np,NMAX,ITMAX
REAL ftol,p(mp,np),y(mp)
PARAMETER (NMAX=20,ITMAX=500)
EXTERNAL funk,amotry
INTEGER i,ih,ihi,ilo,inh,i,j,m,n
REAL rtol,sum,swap,ysave,ytry,psum(NMAX)
iter=0
1  do n=1,ndim
    sum=0.
    do m=1,ndim+1
        sum=sum+p(m,n)
    end do

```

```

    psum(n)=sum
    end do
2  ilo=1
    if (y(1).gt.y(2)) then
        ihi=1
        inhi=2
    else
        ihi=2
        inhi=1
    endif
    do i=1,ndim+1
        if(y(i).le.y(ilo)) ilo=i
        if(y(i).gt.y(ihi)) then
            inhi=ihl
            ihi=i
        else if(y(i).gt.y(inhi)) then
            if(i.ne.ihl) inhi=i
        endif
    end do
    rtol=abs(y(ihl)-y(ilo))
    write(6,*) "tolerance=",ftol,rtol
    print*, "1",y(1),y(2),y(3),y(4)
    if (rtol.lt.ftol) then
        swap=y(1)
        y(1)=y(ilo)
        y(ilo)=swap
    do n=1,ndim
        swap=p(1,n)
        p(1,n)=p(ilo,n)
    end do

```

```

        p(ilo,n)=swap
    end do

    return
endif
if (iter.ge.ITMAX) then
    write(6,*) 'max_ iterations depasse veuillez reprendre avec de nouvelles valeurs
initiales'
    convergence_atteinte=0
    return
end if
iter=iter+2
write(6,*) 'iter= ',iter
ytry=amotry(p,y,psum,mp,np,ndim,funk,ihl,-1.0)
if (ytry.le.y(ilo)) then
    ytry=amotry(p,y,psum,mp,np,ndim,funk,ihl,2.0)
else if (ytry.ge.y(inhl)) then
    ysave=y(ihl)
    ytry=amotry(p,y,psum,mp,np,ndim,funk,ihl,0.5)
    if (ytry.ge.ysave) then
        do i=1,ndim+1
            if(i.ne.ilo)then
                do j=1,ndim
                    psum(j)=0.5*(p(i,j)+p(ilo,j))
                    p(i,j)=psum(j)
                end do
                y(i)=funk(psum)
            endif
        end do
    end if
end do

```

```

        iter=iter+ndim
        goto 1
    endif
else
    iter=iter-1
endif
goto 2
END subroutine amoeba

FUNCTION amotry(p,y,psum,mp,np,ndim,funk,ihl,fac)
INTEGER ihl,mp,ndim,np,NMAX
logical negative
REAL amotry,fac,p(mp,np),psum(np),y(mp),funk
PARAMETER (NMAX=20)
EXTERNAL funk
INTEGER j
REAL fac1,fac2,ytry,ptry(NMAX)
10  fac1=(1.-fac)/ndim
    fac2=fac1-fac
        do j=1,ndim
            ptry(j)=psum(j)*fac1-p(ihl,j)*fac2
        end do
    negative=.false.
    do j=1,ndim
        if(ptry(j).lt.0.) then
            negative=.true.
        end if
    end do
    if(negative) then

```

```
    fac=.5*fac

    goto 10
end if
ytry=funk(ptry)
    if (ytry.lt.y(ihi)) then
        y(ihi)=ytry
        do j=1,ndim
            psum(j)=psum(j)-p(ihi,j)+ptry(j)
            p(ihi,j)=ptry(j)
        end do
    endif
    amotry=ytry
return
END function amotry
end Module
```

```

! -----
!
!fichier principal.f95
!          contient le programme principal
!
!   Created on Tue Jun 14 18:04:50 2005
!   Copyright (c) 2005 Foutlane. All rights reserved.
! -----

use prevision
use initialisation
use lecture
use lecture_resultat
use estimation
use affichage
integer l
call lire_afficher('entrees.txt')
Mo=int(parametres(8))
Io=int(parametres(9))
l=int(parametres(10))
critere=int(parametres(11))
MaxSimulations=int(parametres(12))
precisio=parametres(13)
chai=debut(l)
call lire_resultat('resultat.txt')
if(parametres(1)==1) then
call prevoir(chai,Mo,Io,parametres(2:4))
else
call estimer()
if(convergence_atteinte==0) stop

```

```

end if
call ecrire_polyfinal('polyfinal.txt')
call ecrire_probabi('proba.txt')
call ecrire_moments('moments.txt')
call afficher()
end

! -----
!
!fichier Lecture_resultat.f95
!          contient la procédure de lecture des résultats
!          expérimentaux
!
!   Created on Thu Jun 16 02:37:30 2005
!   Copyright (c) 2005 Foutlane. All rights reserved.

! -----

Module lecture_resultat
use resultat_experimental
contains
subroutine lire_resultat(nom)
    character(len=*):: nom
    integer, parameter:: f=8
    integer :: ios,debut
    integer :: k
    character(len=1)::c
    logical file_existed
    inquire(File=nom,Exist=file_existed)
    if(file_existed==.false.) then

```

```

        print*,"le fichier des resultats n existe pas"
        stop
    end if
    open(unit=f,File=nom,iostat=ios)
    if(ios/=0) then
        print*,"le fichier resultat experimental pas ouvert"
        stop
    end if
    debut=0
1    if(debut==1) then
        print*,":"
        read(unit=f,*) k
        nbre_source=k
        print*,nbre_source
        end if
    if(debut==2) then
        print*,":"
        read(unit=f,*) k
        nbre_observations=k
        print*,k
    end if
    if(debut==3) then
        do j=1,nbre_observations
            read(unit=f,*) tabtemps(j),tabX(j,1:2)
        end do
    end if
    if(debut==4) then

        read(unit=f,*) Ma(1:nbre_source)

```

```

        end if
    debut=debut+1
    do while(debut<5)
        read(unit=f,'(a)',advance='no',eor=1,end=2) c
        write(*,'(a)',advance='no') c        end do
2    print*,"fin de lecture des resultats"
    tabtemps=tabtemps*60
    tabX(:,2)=tabX(:,2)*1e3
    close(f)
end subroutine
end module

```

```

! -----
!fichier Lecture.f95
!          contient la procédure de lecture des
!          paramètres du système réactionnel
!   Created on Wed Jun 15 18:03:49 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
!!! -----

```

```

module lecture
use donneesglobal
contains
subroutine lire_afficher(nom)
    character(len=*):: nom
    integer, parameter:: f=8
    integer :: ios,debut
    real :: k
    character(len=1)::c
    logical file_existed

```

```

inquire(File=nom,Exist=file_existed)
if(file_existed==.false.) then
    print*,"le fichier des donnees n existe pas"
    stop
end if
open(unit=f,File=nom,iostat=ios)
if(ios/=0) then
    print*,"le fichier pas ouvert"
    stop
end if
debut=0

1    if(debut/=0) then
        print*,":"
        read(unit=f,*) k
        parametres(debut)=k
        print*,parametres(debut)

    end if
        debut=debut+1
    do
        read(unit=f,'(a)',advance='no',eor=1,end=2) c
        write(*,'(a)',advance='no') c
    end do
2    print*,"fin de lecture des donnees"
        close(f)
end subroutine lire_afficher
end module lecture

```

```

! -----
!
!fichier Isographe.f95
!
!      contient les procédures de l'algorithme de Cordella
!      pour vérifier l isomorphisme de deux graphes
!
!   Created on Tue Jun 14 16:45:47 2005
!   Copyright (c) 2005 Foutlane. All rights reserved.
! -----

```

Module isomorphisme

```

    use chain
    integer, parameter::MAX_DIM=30
    type ullState
    integer core_len
    integer,Dimension(:),Pointer::core_1,core_2
    type (graph),Pointer::g1,g2
    integer ::n1,n2
    integer*1,Dimension(:,:),Pointer::M
    end type ullState
    contains
    logical function match(g1,g2,pn,c1,c2)
    type (graph),Pointer::g1,g2
    integer,Pointer::pn
    integer,Dimension(:),Pointer::c1,c2
    type (ullState):: s0
    s0=constructeur_etat(g1,g2)
    match=match2(pn,c1,c2,s0)
    end function match

```

```

function match2(pn,c1,c2,s0) result(resultat)
logical::resultat
integer,Pointer::pn
integer,Dimension(:),Pointer::c1,c2
type (ullState):: s0,s1
integer,target::n1,n2
integer,Pointer::pt1,pt2
logical::found
n1=NULL_NODE
n2=NULL_NODE
pt1=>n1
pt2=>n2
if(IsGoal(s0)) then
pn=CoreLen(s0)
resultat=.true.
call destructeur_etat(s0)
return
end if
if(IsDead(s0)) then
resultat=.false.
call destructeur_etat(s0)
return
end if
found=.false.
do while(found==.false. .and. nextPair(s0,pt1,pt2,n1,n2))
if(feasable (s0,n1,n2)) then
s1=constructeur_2etat(s0)
call AddPair(s1,n1,n2)
found=match2(pn,c1,c2,s1)

```

```

end if
end do
resultat=found
call destructeur_etat(s0)
end function match2

```

```

logical function match1(g1,g2)
type(graph),target::g1,g2
type(graph),Pointer::ag1,ag2
integer,dimension(MAX_DIM),target::c1,c2
integer,dimension(:),Pointer::pc1,pc2
integer,pointer::ptr
integer ,target::n1
logical::ans=.false.
ag1=>g1
ag2=>g2
pc1=>c1
pc2=>c2
ptr=>n1
ans=match(ag1,ag2,ptr,pc1,pc2)
if(ans) then
match1=.true.
else
match1=.false.
end if
end function match1
subroutine refine (etat)
type(ullState)::etat
integer i,j,k,l

```

```

do i=etat%core_len,etat%n1
do j=0,etat%n2-1
if(etat%M(i,j)) then
do k=0,etat%core_len-1
l=etat%core_1(k)
if(l==NULL_NODE) then
goto 10
end if
if (((existe_Arc(etat%g1,i,k)/=existe_Arc(etat%g2,j,l)).or.(existe_Arc(etat%g1,k,i)/=existe_Arc(etat%g2,l,j)))) then
etat%M(i,j)=0
exit
end if
10 end do
end if
end do
end do
end subroutine refine

type (ullState) function constructeur_etat(ag1,ag2)
type (graph),Pointer::ag1
type (graph),Pointer::ag2
integer ::i,j
constructeur_etat%g1=>ag1
constructeur_etat%g2=>ag2
constructeur_etat%n1=compteur_Noeuds(ag1)
constructeur_etat%n2=compteur_Noeuds(ag2)
constructeur_etat%core_len=0

allocate(constructeur_etat%core_1(0:constructeur_etat%n1-1))

```

```

allocate(constructeur_etat%core_2(0:constructeur_etat%n2-1))
allocate(constructeur_etat%M(0:constructeur_etat%n1-
1,0:constructeur_etat%n2-1))
constructeur_etat%core_1=NULL_NODE
constructeur_etat%core_2=NULL_NODE

do i=0,constructeur_etat%n1-1
do j=0,constructeur_etat%n2-1
if((compteur_Entrant(i,constructeur_etat%g1)==compteur_Entrant(j,constructeur
_etat%g2))&
.and.(compteur_Sortant(i,constructeur_etat%g1)==compteur_Sortant(j,constructeur_etat
%&
g2)).and.constructeur_etat%g1%groupe(i)==constructeur_etat%g2%groupe(j))then
constructeur_etat%M(i,j)=1
else

constructeur_etat%M(i,j)=0
end if
end do
end do
end function constructeur_etat

```

```

type (ullState) function constructeur_2etat(etat)
type (ullState)::etat
integer ::i,j
constructeur_2etat%g1=>etat%g1
constructeur_2etat%g2=>etat%g2
constructeur_2etat%n1=etat%n1

```

```

constructeur_2etat%n2=etat%n2
constructeur_2etat%core_len=etat%core_len
allocate(constructeur_2etat%core_1(0:constructeur_2etat%n1-1))
allocate(constructeur_2etat%core_2(0:constructeur_2etat%n2-1))
allocate(constructeur_2etat%M(0:constructeur_2etat%n1-
1,0:constructeur_2etat%n2-1))
constructeur_2etat%core_1(:)=etat%core_1(:)
constructeur_2etat%core_2(:)=etat%core_2(:)
constructeur_2etat%M=0
do i=constructeur_2etat%core_len,constructeur_2etat%n1-1
do j=0,constructeur_2etat%n2-1
constructeur_2etat%M(i,j)=etat%M(i,j)
end do
end do
end function constructeur_2etat

```

```

subroutine destructeur_eta( eta)
type(ullState) eta
deallocate(eta%core_1)
deallocate(eta%core_2)
deallocate(eta%M)
end subroutine destructeur_eta

```

```

logical function nextPair(etat,Noeud1,Noeud2,Noeud3,Noeud4)
type (ullState):: etat
integer,Pointer::Noeud1,Noeud2
integer::Noeud3,Noeud4
if(Noeud3==NULL_NODE) then

```

```

Noeud3=etat%core_len
Noeud4=0
else if (Noeud4==NULL_NODE) then
Noeud4=0
else
Noeud4=Noeud4+1
end if
if(Noeud4>=etat%n2) then
Noeud3=noeud3+1
Noeud4=0
end if
if(Noeud3/=etat%core_len) then
nextPair=.false.
return
end if
do while((Noeud4<etat%n2) .and.(etat%M(Noeud3,Noeud4)==0))
Noeud4=Noeud4+1
end do
if(Noeud4<etat%n2) then
Noeud1=Noeud3
Noeud2=noeud4
nextPair=.true.
return
else
nextPair=.false.
end if
end function nextPair
logical function feasible (etat,Noeud1,Noeud2)
type (ullState):: etat

```

```

integer::Noeud1,Noeud2
feasable=(etat%M(Noeud1,Noeud2)/=0)
end function feasable

subroutine AddPair(etat,Noeud1,Noeud2)
type (ullState):: etat
integer::Noeud1,Noeud2,k
etat%core_1(Noeud1)=Noeud2
etat%core_2(Noeud2)=Noeud1
etat%core_len=etat%core_len+1
do k=etat%core_len,etat%n1-1
etat%M(k,Noeud2)=0
end do
call refine(etat)
end subroutine AddPair

subroutine GetCoreSet(etat,c1,c2)
integer,Dimension(:),Pointer::c1,c2
type (ullState):: etat
integer i,j
j=0
do i=0,etat%n1-1
if(etat%core_1(i)/=NULL_NODE) then
c1(j)=i
c2(j)=etat%core_1(i)
j=j+1
end if
end do
end subroutine GetCoreSet

```

```

logical function IsGoal(etat)
type (ullState):: etat
integer i,j
if((etat%core_len/=etat%n1) .or. (etat%core_len/=etat%n2))then
IsGoal=.false.
return
end if
do i=0,etat%n1-1
do j=0,etat%n2-1
if((etat%g1%matrice(i,j)/=etat%g2%matrice(etat%core_1(i),etat%core_1(j))&
).or.(etat%g1%matrice_etat(i,j)/=etat%g2%matrice_etat(etat%core_1(i),etat%core_1(j))))then
IsGoal=.false.
return
end if
end do
end do
IsGoal=.true.
end function IsGoal

logical function IsDead(etat)
type (ullState):: etat
integer i,j,u
if(etat%n1/=etat%n2) then
IsDead=.true.
return
end if
do i=etat%core_len,etat%n1-1
do j=0,etat%n2-1

```

```

if(etat%M(i,j)/=0) then
goto 100
end if
end do
IsDead=.true.
return
100 u=4
end do
IsDead=.false.
end function IsDead

```

```

integer function CoreLen(etat)
type (ullState):: etat
CoreLen=etat%core_len
end function CoreLen
logical function Vrai(etat)
type (ullState):: etat
integer i,j
do i=0,etat%n1-1
do j=0,etat%n2-1
if(etat%g1%matrice(i,j)/=etat%g2%matrice(etat%core_1(i),etat%core_1(j))) then
Vrai=.false.
return
end if
end do
end do
Vrai=.true.
end function Vrai

```

```

end Module isomorphisme

```

```

! -----
! fichier initialisation.f95
!
!      contient les procédures et les déclarations des variables
!      nécessaires pour définir l'état initial du système
!      réactionnel.
!
!   Created on Tue Jun 14 18:33:46 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module initialisation
use type_de_donnees
type(initialiseur) ::initia,initia2
type(chaine):: chaine_glycerol,chaine_Benzyl,chaine_Butanediol,chaine_Pentaerythritol
contains

    subroutine initiaBenzylAlcol(initia)
        type(initialiseur) ::initia
        initia%nombre_noyaux=2
        initia%tab_noyaux(1)=constructeur_noyau(7,8,1,0)
        initia%tab_noyaux(2)=constructeur_noyau(0,1,0,0)

        initia%nombre_liaison_variables=1
        initia%tab_coord_variables(1)%ligne=1
        initia%tab_coord_variables(1)%colonne=2
    end subroutine initiaBenzylAlcol

    subroutine initiaButanediol(initia)
        type(initialiseur) ::initia
        initia%nombre_noyaux=3
        initia%tab_noyaux(1)=constructeur_noyau(4,8,2,0)
        initia%tab_noyaux(2)=constructeur_noyau(0,1,0,0)

```

```

        initia%tab_noyaux(3)=constructeur_noyau(0,1,0,0)
        initia%nombre_liaison_variables=2
        initia%tab_coord_variables(1)%ligne=1
        initia%tab_coord_variables(1)%colonne=2
        initia%tab_coord_variables(2)%ligne=1
        initia%tab_coord_variables(2)%colonne=3
    end subroutine

```

```

subroutine initiaPentaerythritol(initia)
    type(initialiseur) ::initia
    initia%nombre_noyaux=5
    initia%tab_noyaux(1)=constructeur_noyau(4,8,2,0)
    initia%tab_noyaux(2)=constructeur_noyau(0,1,0,0)
    initia%tab_noyaux(3)=constructeur_noyau(0,1,0,0)
    initia%tab_noyaux(4)=constructeur_noyau(0,1,0,0)
    initia%tab_noyaux(5)=constructeur_noyau(0,1,0,0)
    initia%nombre_liaison_variables=4
    initia%tab_coord_variables(1)%ligne=1
    initia%tab_coord_variables(1)%colonne=2
    initia%tab_coord_variables(2)%ligne=1
    initia%tab_coord_variables(2)%colonne=3
    initia%tab_coord_variables(3)%ligne=1
    initia%tab_coord_variables(3)%colonne=4
    initia%tab_coord_variables(4)%ligne=1
    initia%tab_coord_variables(4)%colonne=5
end subroutine

```

```

type(chaine) function debut(l)
    integer k

```

```

type(chaine) chain
allocate(tab_initialiseurs(4))
if(l==1) then
    call initiaBenzylAlcol(initia)
    chaine_Benzyl%type_initialiseur=1
    chaine_Benzyl%nombre_liaison=1
    chaine_Benzyl%frequence=1
    chaine_Benzyl%tab_liaison(1)%indice_coord=1
    chaine_Benzyl%tab_liaison(1)%etat=0
    chaine_Benzyl%tab_liaison(1)%nbre_unite_repetitif=0
    chain=chaine_Benzyl
else if(l==2) then
    call initiaButanediol(initia)
    chaine_Butanediol%type_initialiseur=1
    chaine_Butanediol%nombre_liaison=2
    chaine_Butanediol%frequence=1
    do k=1,2
        chaine_Butanediol%tab_liaison(k)%indice_coord=k
        chaine_Butanediol%tab_liaison(k)%etat=0

chaine_Butanediol%tab_liaison(k)%nbre_unite_repetitif=0
    end do
    chain=chaine_Butanediol
else if(l==3) then
    call initiaPentaerythritol(initia)
    chaine_Pentaerythritol%type_initialiseur=1
    chaine_Pentaerythritol%nombre_liaison=4
    chaine_Pentaerythritol%frequence=1
    do k=1,4

```

```

chaine_Pentaerythritol%tab_liaison(k)%indice_coord=k
chaine_Pentaerythritol%tab_liaison(k)%etat=0

chaine_Pentaerythritol%tab_liaison(k)%nbre_unite_repetitif=0
end do
chain=chaine_Pentaerythritol

end if
tab_initialiseurs(1)=initia
debut=chain
end function
end Module

! -----
!fichier graphiques.f95
!      contient les procédures pour réaliser les graphes de:
!      -probabilités en fonction de la conversion
!      -conversion en fonction du temps
!      -moments en fonction de la conversion
!      - de Mw en fonction du temps
!      Created on Tue Jun 14 17:58:03 2005
!      Copyright (c) 2005 Foutlane. All rights reserved.
! -----

Module graphique
contains
      subroutine subprobaconversion( prob1total, prob2total, prob3total, prob4total,
converstotal, replications)
      real,Dimension(:,:): prob1total, prob2total, prob3total, prob4total,
converstotal

```

```

integer h,j,k,replications
real,Dimension(100)::convers,prob1moyen,prob2moyen,prob3moyen,
prob4moyen
h=100
do k=1,replications
  do j=1,100
    if(converstotal(k,j)==-1) then
      if(j-1<h) then
        h=j-1
      end if
    end if
  end do
end do

do k=1,h
  prob1moyen(k)=sum(prob1total(1:replications,k))/replications

  prob2moyen(k)=sum(prob2total(1:replications,k))/replications

  prob3moyen(k)=sum(prob3total(1:replications,k))/replications

  prob4moyen(k)=sum(prob4total(1:replications,k))/replications
end do
convers=/(k*0.02,k=0,h-1/)
call plenv(0.0,1.0,0.0,1.0,0,0)
call plcol(1)
call plline(h,convers,prob1moyen)

```

```

        call plcol(6)
        call ppline(h,convers,prob2moyen)
        call plcol(9)
        call ppline(h,convers,prob3moyen)
        call plcol(15)
        call ppline(h,convers,prob4moyen)
        call plcol(2)
        call pllab('#frconversion (%)','#frprobabilities','#frdistribution des
probabilites')

    end subroutine subprobaconversion

    subroutine subconversiontemps (nbpnts,convers, tempsmoyen,observations,
dataexp,tabtemps)
        integer nbpnts,observations
        real,Dimension(:,)::dataexp
        real,Dimension(100)::convers
        real,Dimension(100) ::tempsmoyen,tabtemps
        real maxima
        maxima=1.1*min(maxval(tempsmoyen(1:nbpnts)),maxval(tabtemps(1:observat
ions)))

        call plenv(0.0,maxima,0.0,1.0,0,0)
        call plcol(15)
        call ppline(nbpnts,tempsmoyen(:),convers)
        call plcol(6)
        call
ppline(observations,tabtemps(1:observations),dataexp(1:observations,1))
        call plcol(2)
        call pllab('#frtemps','#frConversion(',')','#frConversion fct du temps ')

```

```

end subroutine subconversiontemps

subroutine submomentsconversion (Mntotal,Mwttotal, Mzttotal, tempsmoyen,
converstotal, replications)
  integer h,j,k,replications
  real,Dimension(:,:):Mntotal,Mwttotal,Mzttotal,converstotal
  real,Dimension(100):: Mnmoyen,Mwmoyen,Mzmoyen
  real,Dimension(100):: tempsmoyen
  real maxi,maxima
  h=100
  do k=1,replications
    do j=1,100
      if(converstotal(k,j)=-1) then
        if(j-1<h) then
          h=j-1
        end if
        exit
      end if
    end do
  end do

  do k=1,h
    Mnmoyen(k)=sum(Mntotal(1:replications,k))/replications

    Mwmoyen(k)=sum(Mwttotal(1:replications,k))/replications

    Mzmoyen(k)=sum(Mzttotal(1:replications,k))/replications
  end do

```

```

maxi=1.1*maxval(tempsmoyen(1:h))

maxima=1.1*max(maxval(Mnmoyen(1:h)),maxval(Mwmoyen(1:h)),maxval(Mzmoyen(1:h)))
call plenv(0.0,maxi,0.0,maxima,0,0)
call plcol(1)
call plline(h,tempsmoyen,Mnmoyen)
call plcol(6)
call plline(h,tempsmoyen,Mwmoyen)
call plcol(9)
call plline(h,tempsmoyen,Mzmoyen)
call plcol(2)
call pllabb('#frtemps','#frmoments','#frevolution des moments ')

end subroutine submomentsconversion

subroutine subMw (nbpoints, Mwmoyen, tempsmoyen, observations,
dataexp,tabtemps)
integer nbpoints,observations
real,Dimension(:,)::dataexp
real,Dimension(100):: Mwmoyen
real,Dimension(100):: tabtemps,tempsmoyen
real maxi,maxima
maxi=1.1*min(maxval(tempsmoyen(1:nbpoints)),maxval(tabtemps(1:observations)))

maxima=1.1*max(maxval(Mwmoyen(1:nbpoints)),maxval(dataexp(1:observations,2)))

call plenv(0.0,maxi,0.0,maxima,0,0)
call plcol(6)

```

```

        call ppline(nbpoints,tempsmoyen,Mwmoyen)
        call plcol(9)
        call ppline(observations,tabtemps,dataexp(:,2))
        call plcol(2)
        call pllab('#frtemps','#frmoments','#frevolution de Mwsim et Mwexp')
    end subroutine subMw
end Module graph

```

```

! -----
!fichier global.f95
!
!      contient les données globales du système
!      et du vecteur des paramètres du choix de
!      l'utilisateur
!
!   Created on Wed Jun 15 18:14:12 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

```

```

Module donneesglobal
use type_de_donnees
    integer Io,Mo
    type(chaine) chai
    real parametres(14)
end Module donneesglobal

```

! -----

!fichier evenements.f95

! contient les procédures correspondant aux
! réactions qui se produisent dans le système:
! -initiation
! -propagation
! -terminaison
! des ajustements que subissent le tableau des sites
! initialisés et ceux désactivés.
! et de recherche des sites de réactions

! Created on Tue Jun 14 17:55:42 2005

! Copyright (c) 2005foutlane. All rights reserved.

! -----

Module operations

use comparer

contains

 subroutine

subinitiation(tab_chaines,nombre_chaines_différentes,nombre_chaines,lo,chai)

 type(chaine),Dimension(:),Pointer::tab_chaines

 type(chaine)::chai

 logical bb

 real u

 integer kk,j,nombre_chaines_différentes,nombre_chaines,lo

 bb=.false.

 if(nombre_chaines==0) then

 nombre_chaines=nombre_chaines+1

```

nombre_chaines_différentes=nombre_chaines_différentes+1

tab_chaines(nombre_chaines_différentes)=chai

else
  if(nombre_chaines<Io .and.(nombre_chaines/=0)) then

    nombre_chaines=nombre_chaines+1
    u=Masse_Molaire(chai,Masse_Monomere)
    j=1

    do,
      while((Masse_Molaire(tab_chaines(j),Masse_Monomere)<=u).and.(j<nombre_chaines_
différentes))

        bb=sont_identiques(tab_chaines(j),chai,tab_initialiseurs)

        if(bb)then

          tab_chaines(j)%fréquence=tab_chaines(j)%fréquence+1
          exit
        end if
        j=j+1
      end do
      if(j==nombre_chaines_différentes) then

        bb=sont_identiques(tab_chaines(j),chai,tab_initialiseurs)

        if(bb)then

```

```

tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
    end if
end if
if(.not.bb) then
    do kk=nombre_chaines_différentes,1,-1
        tab_chaines(kk+1)=tab_chaines(kk)
    end do
    tab_chaines(1)=chai

nombre_chaines_différentes=nombre_chaines_différentes+1

    end if
end if
end if

end subroutine subinitiation
subroutine subtransfert (tab_chaines,k,j,nombre_chaines_différentes,sortie)
    integer k,j,nombre_chaines_différentes,sortie
    type(chaine),Dimension(:),Pointer::tab_chaines
    type(chaine)::chai2
    real u
    type(site)::cite
    logical bb
    bb=.false.
    if(tab_chaines(k)%tab_liaison(j)%etat/=1) then

        cite=site_vide(k,tab_chaines,1,nombre_chaines_différentes)
        if(cite%numero==-1) then

```

```

        print*,"pas de site pour terminaison"
        sortie=1
        return
    else
        k=cite%numero
        j=cite%branche
    end if
end if
tab_chaines(k)%frequence=tab_chaines(k)%frequence-1
chai2=tab_chaines(k)
chai2%frequence=1
chai2%tab_liaison(j)%etat=-1
u=Masse_Molaire(chai2,Masse_Monomere)
if(tab_chaines(k)%frequence==0) then
    do j=k,nombre_chaines_différentes-1
        tab_chaines(j)=tab_chaines(j+1)
    end do
    nombre_chaines_différentes=nombre_chaines_différentes-1
end if
j=k
do,
while((Masse_Molaire(tab_chaines(j),Masse_Monomere)>=u).and.(j>1))
    j=j-1
end do

do,
while((Masse_Molaire(tab_chaines(j),Masse_Monomere)<=u).and.(j<nombre_chaines_
différentes))
    bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)

```

```

        if(bb)then
            tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
            exit
        end if
        j=j+1
    end do
    if(j==nombre_chaines_différentes) then
        bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)
        if(bb)then
            tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
        else

nombre_chaines_différentes=nombre_chaines_différentes+1
            if(u>Masse_Molaire(tab_chaines(j),Masse_Monomere))
then
                tab_chaines(nombre_chaines_différentes)=chai2
            else

tab_chaines(nombre_chaines_différentes)=tab_chaines(j)
                tab_chaines(j)=chai2
            end if
        end if
    else

        if(.not.bb) then
            do kk=nombre_chaines_différentes,j,-1
                tab_chaines(kk+1)=tab_chaines(kk)
            end do
            tab_chaines(j)=chai2
        end if
    end if
end if

```

```

nombre_chaines_différentes=nombre_chaines_différentes+1
    end if
end if
end subroutine subtransfert

subroutine subpropagation(tab_chaines,k,j,nombre_chaines_différentes,sortie&
,nombre_monomeres)
    integer k,j,nombre_chaines_différentes,sortie
    type(chaine),Dimension(:),Pointer::tab_chaines
    type(chaine)::chai2
    real u
    type(site)::cite
    logical bb
    bb=.false.
    if(.not.(tab_chaines(k)%tab_liaison(j)%etat==1)) then
        cite=site_vide(k,tab_chaines,1,nombre_chaines_différentes)
        if(cite%numero==-1) then
            print*,"pas de site pour propagation"
            sortie=1

            return
        else
            k=cite%numero
            j=cite%branche
        end if
    end if
end if
nombre_monomeres=nombre_monomeres-1
tab_chaines(k)%frequence=tab_chaines(k)%frequence-1

```

```

chai2=tab_chaines(k)
chai2%frequence=1

chai2%tab_liaison(j)%nbre_unite_repetitif=chai2%tab_liaison(j)&
    %nbre_unite_repetitif + 2

if(tab_chaines(k)%frequence==0) then
    do j=k,nombre_chaines_différentes-1
        tab_chaines(j)=tab_chaines(j+1)
    end do
    nombre_chaines_différentes=nombre_chaines_différentes-
1
end if

j=k
u=Masse_Molaire(chai2,Masse_Monomere)
do,
while((j<nombre_chaines_différentes).and.(Masse_Molaire(tab_chaines(j),Masse_Mono
mere)<u))
    j=j+1
end do
if(j==nombre_chaines_différentes) then
    bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)
    if(bb) then

tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
    else

nombre_chaines_différentes=nombre_chaines_différentes+1

```

```

if(u>Masse_Molaire(tab_chaines(j),Masse_Monomere)) then

tab_chaines(nombre_chaines_différentes)=chai2
    else
        tab_chaines(nombre_chaines_différentes)=tab_chaines(j)
        tab_chaines(j)=chai2
    end if
end if
else
do,
while((Masse_Molaire(tab_chaines(j),Masse_Monomere)<=u).and.(j<nombre_chaines_
différentes))

bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)
    if(bb)then
        tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
        exit
    end if
    j=j+1
end do
if(.not.bb) then
do kk=nombre_chaines_différentes,j,-1
        tab_chaines(kk+1)=tab_chaines(kk)
end do
tab_chaines(j)=chai2

```

```

nombre_chaines_différentes=nombre_chaines_différentes+1
    end if
end if
end subroutine subpropagation

```

```

subroutine subinitiation2(tab_chaines,nombre_chaines_différentes,k,diffe)
integer k,j,nombre_chaines_différentes,diffe,h,l,gg
    type(chaine),Dimension(:),Pointer::tab_chaines
    type(chaine)::chai2
    real u
    logical bb
    bb=.false.
    gg=0
    h=diffe/tab_chaines(k)%frequence
    tab_chaines(k)%frequence=tab_chaines(k)%frequence-1
    chai2=tab_chaines(k)
    chai2%frequence=1
    do l=1,chai2%nombre_liaison
    if(chai2%tab_liaison(l)%etat==0) then
        gg=gg+1
        if(gg==h) then
            exit
        end if
    end if
    end do
    chai2%tab_liaison(l)%etat=1
    u=Masse_Molaire(chai2,Masse_Monomere)
    if(tab_chaines(k)%frequence==0) then

```

```

do j=k,nombre_chaines_différentes-1
    tab_chaines(j)=tab_chaines(j+1)
end do
nombre_chaines_différentes=nombre_chaines_différentes-1
end if
j=k
do,
while((Masse_Molaire(tab_chaines(j),Masse_Monomere)>=u).and.(j>1))
    j=j-1
end do
do,
while((Masse_Molaire(tab_chaines(j),Masse_Monomere)<=u).and.(j<nombre_chaines_
différentes))
    bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)
    if(bb)then
        tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
        exit
    end if
    j=j+1
end do
if(j==nombre_chaines_différentes) then
    bb=sont_identiques(tab_chaines(j),chai2,tab_initialiseurs)
    if(bb)then
        tab_chaines(j)%frequence=tab_chaines(j)%frequence+1
    else

nombre_chaines_différentes=nombre_chaines_différentes+1
        if(u>Masse_Molaire(tab_chaines(j),Masse_Monomere))
then

```

```

        tab_chaines(nombre_chaines_différentes)=chai2
    else
        tab_chaines(nombre_chaines_différentes)=tab_chaines(j)
        tab_chaines(j)=chai2
    end if
end if
else
    if(.not.bb) then
        do kk=nombre_chaines_différentes,j,-1
            tab_chaines(kk+1)=tab_chaines(kk)
        end do
        tab_chaines(j)=chai2
    end if
    nombre_chaines_différentes=nombre_chaines_différentes+1
end if
end subroutine

```

```

subroutine subajustement(tab,tableau,nombre)
    type(chaine),Dimension(:),Pointer::tableau
    integer,Dimension(:),Pointer:: tab
    integer nombre,i,k,g,d
    if(nombre>0) then
        tab(1)=0
        d=tableau(1)%nombre_liaison
        g=0
        do k=1,d
            if(tableau(1)%tab_liaison(k)%etat==1) then

```

```

        g=g+1
    end if
end do
tab(1)=g*tableau(1)%frequence
do i=2,nombre
    g=0
    do k=1,d
        if(tableau(i)%tab_liaison(k)%etat==1) then
            g=g+1
        end if
    end do
    tab(i)=tab(i-1)+g*tableau(i)%frequence
end do
end if
end subroutine subajustement

```

```

subroutine subajustementIni(tab,tableau,nombre)
    type(chaine),Dimension(:),Pointer::tableau
    integer,Dimension(:),Pointer:: tab
    integer nombre,i,k,g,d
    if(nombre>0) then
        tab(1)=0
        d=tableau(1)%nombre_liaison
        g=0
        do k=1,d
            if(tableau(1)%tab_liaison(k)%etat==0) then
                g=g+1
            end if
        end do
    end do

```

```

    tab(1)=g*tableau(1)%frequence
    do i=2,nombre
        g=0
        do k=1,d
            if(tableau(i)%tab_liaison(k)%etat==0) then
                g=g+1
            end if
        end do

        tab(i)=tab(i-1)+g*tableau(i)%frequence
    end do
end if
end subroutine

integer function case_reaction(tableau,limite,fin)
    integer,Dimension(:),Pointer::tableau
    integer fin,k,limite
    do k=1,fin
        if(tableau(k)>= limite) then
            case_reaction=k
            exit
        end if
    end do
end function case_reaction

type(site) function site_vide (depart,tableau,debut,fin)
    integer depart,debut,fin
    type(chaine),Dimension(:),Pointer::tableau
    integer indice,indice_croissant,indice_decroissant
    integer i,d

```

```

logical trouve,boolsup
trouve=.false.
boolsup=.true.
indice=depart
indice_croissant=depart+1
indice_decroissant=depart -1
d=tableau(indice)%nombre_liaison
do i=1,d
    if(tableau(indice)%tab_liaison(i)%etat==1) then
        site_vide%numero=indice
        site_vide%branche=i
        trouve=.true.
        exit
    end if
end do
do,while(.not.trouve)
    if(indice_croissant<=fin) then
        do i=1,d

if(tableau(indice_croissant)%tab_liaison(i)%etat==1) then
            site_vide%numero=indice_croissant
            site_vide%branche=i
            trouve=.true.
            exit
        end if
    end do
end if

    if(indice_decroissant>= debut) then

```

```

do i=1,d

if(tableau(indice_decroissant)%tab_liaison(i)%etat==1) then

site_vide%numero=indice_decroissant

site_vide%branche=i
trouve=.true.
exit
end if
end do
end if

if(indice_croissant>fin .and. indice_decroissant<debut) then
site_vide%numero=-1
site_vide%branche=-1
exit

end if
indice_croissant=indice_croissant+1
indice_decroissant=indice_decroissant-1
end do
end function site_vide
end module operations

```

```

! -----
!fichier estimation.f95
!          contient la procédure d'estimation des coefficients
!          stochastiques
!   Created on Wed Jun 15 21:30:23 2005
!   Copyright (c) 2005 foutlane. All rights reserved.
! -----

Module estimation
use Minimisation
contains
subroutine estimer()
    implicit none
    integer iter,i,j,ios
    real p(4,3),y(4),e(3,3),x(3),ftol,lambda(3)
    e(1,1)=1
    e(1,2)=0
    e(1,3)=0
    e(2,1)=0
    e(2,2)=1
    e(2,3)=0
    e(3,1)=0
    e(3,2)=0
    e(3,3)=1
    open(unit=14,File='optimisation.txt',iostat=ios)
    if(ios/=0) then
        print*, "le fichier optimisation pas ouvert"
    end if
    do i=1,3
        p(1,i)=parametres(i+1)

```

```

        lambda(i)=parametres(i+4)
    end do

    do i=1,3
        do j=1,3
            p(i+1,j)=p(1,j)+lambda(i)*e(i,j)
        end do
    end do

    write(6,*) "on debute par le calcul de f aux points du tetrahedre"
    write(14,*) "on debute par le calcul de f aux points du tetrahedre"

    do i=1,4
        x(1)=p(i,1)
        x(2)=p(i,2)
        x(3)=p(i,3)
        y(i)=funk2(x)
        write(6,*) "les f correspondants",(x(j),j=1,3),y(i)
        write(14,*) "les f correspondants",(x(j),j=1,3),y(i)
    end do

    write(14,*) "optimisation initiee"
    write(6,*) "optimisation initiee"

    ftol=1.e-2
    write(6,*) "tolerance=",ftol

    call amoeba(p,y,4,3,3,ftol,funk2,iter,amotry)
    if(convergence_atteinte==1) then
        do j=1,4
            write(14,*) "le point",j," du prisme de convergence ",(p(j,i),i=1,3)
        end do
    end if
end do

```

```

        write(14,*) "le point de convergence p",(p(1,i),i=1,3)
        write(14,*) "iterations faites ",iter
        write(6,*) "le point de convergence p",(p(1,i),i=1,3)
        write(6,*) "iterations faites ",iter
        y(1)=funkt2(p(1,:))
    else
        write(14,*) " pas de convergence atteinte reprendre avec de nouveaux
parametres"
    end if
    close(14)
end subroutine estimer
end Module

! -----
!
! fichier comparaison.f95
!         contient la procédure qui compare deux chaînes
!
!   Created on Wed Jun 15 11:34:15 2005
!   Copyright (c) 2005 Foutlane. All rights reserved.
! -----

Module comparer
use isomorphisme
use type_de_donnees
contains
logical function sont_identiques (chaineA,chaineB,tableau)
type(initialiseur), Dimension(:),Pointer ::tableau
type (graph),target:: g,g3
type(chaine) chaineA,chaineB
type (initialiseur) initA,initB

```

```

integer k,j
integer*1,Dimension(:),Pointer::tableau1,tableau2
integer,Dimension(:,:),Pointer:: matriceA,matriceB
integer,Dimension(:,:),Pointer:: matrice_etat1,matrice_etat2
if(Masse_Molaire(chaineA,Masse_Monomere)/=Masse_Molaire(chaineB,Masse_Monomere)) then
sont_identiques=.false.
return
else
initA= tableau(chaineA%type_initialiseur)
initB= tableau(chaineB%type_initialiseur)
if (initA%nombre_noyaux/=initB%nombre_noyaux) then
sont_identiques=.false.
return
end if
allocate(matriceA(initA%nombre_noyaux,initA%nombre_noyaux))
allocate(matriceB(initB%nombre_noyaux,initB%nombre_noyaux))
allocate(matrice_etat1(initA%nombre_noyaux,initA%nombre_noyaux))
allocate(matrice_etat2(initB%nombre_noyaux,initB%nombre_noyaux))
allocate(tableau1(initA%nombre_noyaux))
allocate(tableau2(initA%nombre_noyaux))
matriceA=0
matriceB=0
matrice_etat1=-2
matrice_etat2=-2
do k=1, initA%nombre_noyaux
tableau1(k)=k
end do
do k=1, initA%nombre_noyaux

```

```

do j=k+1, initA%nombre_noyaux
  if (StEgales(initA%tab_noyaux(j),initA%tab_noyaux(k))) then
    tableau1(j)=tableau1(k)
  end if
end do
end do

do k=1, initB%nombre_noyaux
  tableau2(k)=k+ initB%nombre_noyaux
end do

do k=1, initA%nombre_noyaux
  do j=1, initB%nombre_noyaux
    if (StEgales(initA%tab_noyaux(k),initB%tab_noyaux(j))) then
      tableau2(j)=tableau1(k)
    end if
  end do
end do

do k=1, initA%nombre_liaison_fixes
  matriceA(initA%tab_coord_fixes(k)%ligne,initA%tab_coord_fixes(k)%colonne)=1
  matriceA(initA%tab_coord_fixes(k)%colonne,initA%tab_coord_fixes(k)%ligne)=1
end do

do k=1, initB%nombre_liaison_fixes
  matriceB(initB%tab_coord_fixes(k)%ligne,initB%tab_coord_fixes(k)%colonne)=1
  matriceB(initB%tab_coord_fixes(k)%colonne,initB%tab_coord_fixes(k)%ligne)=1
end do

do k=1, initA%nombre_liaison_variables
  matriceA(initA%tab_coord_variables(k)%ligne,initA%tab_coord_variables(k)%colonne
)=&

```

```

chaîneA%tab_liaison(k)%nbre_unite_repetitif
matriceA(initA%tab_coord_variables(k)%colonne,initA%tab_coord_variables(k)%ligne
)=&
chaîneA%tab_liaison(k)%nbre_unite_repetitif
matrice_etat1(initA%tab_coord_variables(k)%ligne,initA%tab_coord_variables(k)%col
onne)=&
chaîneA%tab_liaison(k)%etat
matrice_etat1(initA%tab_coord_variables(k)%colonne,initA%tab_coord_variables(k)%l
igne)=&
chaîneA%tab_liaison(k)%etat
end do
do k=1, initB%nombre_liaison_variables
matriceB(initB%tab_coord_variables(k)%ligne,initB%tab_coord_variables(k)%colonne)
=&
chaîneB%tab_liaison(k)%nbre_unite_repetitif
matriceB(initB%tab_coord_variables(k)%colonne,initB%tab_coord_variables(k)%ligne)
=&
chaîneB%tab_liaison(k)%nbre_unite_repetitif
matrice_etat2(initB%tab_coord_variables(k)%ligne,initB%tab_coord_variables(k)%colo
nne)=&
chaîneB%tab_liaison(k)%etat
matrice_etat2(initB%tab_coord_variables(k)%colonne,initB%tab_coord_variables(k)%li
gne)=&
chaîneB%tab_liaison(k)%etat
end do
g=constructeur_graph(matriceA,initA%nombre_noyaux,tableau1,matrice_etat1)
g3=constructeur_graph(matriceB,initB%nombre_noyaux,tableau2,matrice_etat2)
sont_identiques=match1(g,g3)
deallocate(matriceA)

```

```

deallocate(matriceB)
deallocate(tableau1)
deallocate(tableau2)
deallocate(matrice_etat1)
deallocate(matrice_etat2)
call destructeur(g)
call destructeur(g3)
end if
end function sont_identiques
end Module

! -----
!
! fichier affichage.f95
!
!         contient la procédure qui affiche les graphiques de:
!
!             -probabilités des événements
!             -le moment Mw
!             -les trois premiers Moments
!             -la conversion
!             -la distribution massique
!
!         et les procédures d'écriture des probabilités et des moments
!   Created on Wed Jun 15 13:46:51 2005
!   Copyright (c) 2005foutlane. All rights reserved.
!   -----

Module affichage
use graph
use resultat_simulation
use resultat_experimental
use type_de_donnees
contains

```

```

subroutine afficher()
    real tab(60)
    integer h
    call plsdev("win")
        call plssub(2, 3)
        call plinit()
call subprobaconversion (prob1total, prob2total, prob3total,prob4total, converstotal,
replications)
call subMw(nbpoints_simu,Moment,tempsmoyen,nbre_observations,tabX,tabtemps)
call submomentsconversion (Mntotal, Mwtotal, Mztotal, tempsmoyen, converstotal,
replications)
call subconversiontemps (nbpoints_simu, convers, tempsmoyen, nbre_observations,
tabX, tabtemps)
    tab=/(h*maxi/60,h=0,59)/
    call plcol(6)
    call plenv(0.0,maxi*1.1,0.0,maxval(tableau)*1.1,0,0)
    call plbin(size(tableau),tab ,tableau, 0)
    call plcol(6)
    call pllab('#frMasse(g/mol)', '#frnombre n', '#frdistribution moyenne des masses ')
call plend
end subroutine

subroutine ecrire_fichier_Sources(nom)
    character(len=*):: nom
    integer, parameter:: f=10
    integer :: ios,k
    open(unit=f,File=nom,iostat=ios)
    if(ios/=0) then
        print*,"le fichier pour ecriture pas ouvert"
        stop
    
```

```

end if
write(10,*) "Resultat de la simulation pour la conversion et Mw:"
write(10,*)" temps(s)  conversion %   Mw (g/mol)"
do k=1,nbpoints_simu
write(10,'(F10.1,F15.2,F15.1)') tempsmoyen(k),convers(k),Moment(k)
end do
close(f)
end subroutine

subroutine ecrire_fichier_Probabilites(nom)
character(len=*):: nom
integer :: ios,k
open(unit=12,File=nom,iostat=ios)
if(ios/=0) then
print*,"le fichier pour ecriture pas ouvert"
stop
end if
write(12,*) "Resultat de l estimation pour probabilites:"
write(12,*)" temps(s)  probabilite1  probabilite2  probabilite3  probabilite4"
do k=1,nbpoints_simu
write(12,'(F10.1,F15.6,F15.6,F15.6,F15.6)') tempsmoyen(k),
prob1moyen(k),prob2moyen(k),prob3moyen(k),prob4moyen(k)
end do
close(12)
end subroutine

subroutine ecrire_polyfinal(nom)
character(len=*):: nom
integer :: ios,k,j

```

```

open(unit=20,File=nom,iostat=ios)
if(ios/=0) then
    print*,"le fichier pour ecriture composition finale pas ouvert"
    stop
end if
write(20,*) "composition finale:"
do k=1, replications
    write(20,*) "numero de la simulation: ",k
    write(20,*) "nombre d especes differents: ",elements_differents(k)
    do j=1,elements_differents(k)
        write(20,*) j, "element de frequence",tab_chaines2(k)%tabchaines(j)%frequence
        write(20,*) " masse ", Masse_Molaire (tab_chaines2(k)%tabchaines(j),
Masse_Monomere)
    end do
    write(20,*) "-----"
end do
close(20)
end subroutine

subroutine ecrire_probabi(nom)
    character(len=*):: nom
    integer :: ios,k,j
    open(unit=24,File=nom,iostat=ios)
    if(ios/=0) then
        print*,"le fichier pour ecriture composition finale pas ouvert"
        stop
    end if
    write(24,*) "probabilites des evenements"
    do k=1, replications

```

```

write(24,*) "numero de la simulation: ",k
write(24,*)" temps(s)  probabilite1  probabilite2  probabilite3  probabilite4"
j=1
do while(converstotal(k,j)/=-1)
write(24,'(F10.1,F15.6,F15.6,F15.6,F15.6)') tempstotal(k,j),
prob1total(k,j),prob2total(k,j),prob3total(k,j),prob4total(k,j)
j=j+1
end do
write(24,*) "-----"
end do
close(24)
end subroutine

subroutine ecrire_moments(nom)
character(len=*):: nom
integer :: ios,k,j
open(unit=26,File=nom,iostat=ios)
if(ios/=0) then
print*, "le fichier pour ecriture composition finale pas ouvert"
stop
end if
write(26,*) "evolution des moments"
do k=1, replications
write(26,*) "numero de la simulation: ",k
write(26,*)" temps(s)  moment1  moment2  moment3 "
j=1
do while(converstotal(k,j)/=-1)
write(26,'(F10.1,F15.2,F15.2,F15.2)') tempstotal(k,j),
Mntotal(k,j),Mwttotal(k,j),Mzttotal(k,j)
j=j+1

```

```

        end do
        write(26,*) "-----"
        end do
        close(26)
end subroutine

```

```

end Module

```

```

! -----
!
! fichier chaine.f95
!          contient les définitions et les procédures de manipulation d'un graphe
!   Created on Tue Jun 14 17:15:23 2005
!   Copyright (c) 2005foutlane. All rights reserved.
! -----

```

```

Module chain

```

```

integer, parameter::NULL_NODE=-1
Type ptr_Node
integer*2,Dimension(:),Pointer ::aa
end Type ptr_Node
Type graph
integer*2 n
integer,Dimension(:,:),Pointer:: matrice
integer,Dimension(:,:),Pointer:: matrice_etat
integer,Dimension(:),Pointer:: in_count
integer,Dimension(:),Pointer:: out_count
type(ptr_Node),Dimension(:),Pointer:: entrant
type(ptr_Node),Dimension(:),Pointer:: sortant
integer*1,Dimension(:),Pointer::groupe

```

```

end type graph
contains
subroutine destructeur_Node (Nod)
type(ptr_Node) Nod
deallocate(Nod%aa)
end subroutine destructeur_Node
type (graph) function constructeur_graph(mat,x,tab,matrice_etat)
integer :: x,i,j,k
integer,allocatable,Dimension(:)::donnee
integer,Dimension(:,:)::mat,matrice_etat
integer*1,Dimension(:)::tab
character(len=10)::valeur
allocate(constructeur_graph%entrant(0:50))
allocate(constructeur_graph%sortant(0:50))
constructeur_graph%n=x
allocate(constructeur_graph%matrice(0:x-1,0:x-1))
allocate(constructeur_graph%matrice_etat(0:x-1,0:x-1))
allocate(constructeur_graph%in_count(0:x-1))
allocate(constructeur_graph%out_count(0:x-1))
allocate(constructeur_graph%groupe(0:x-1))
allocate(donnee(x))
do i=0,x-1
allocate(constructeur_graph%sortant(i)%aa(0:x))
allocate(constructeur_graph%entrant(i)%aa(0:x))
end do
constructeur_graph%in_count=0
constructeur_graph%out_count=0

do i=0,x-1

```

```

donnee(:)=mat(i+1,:);
do j=0,x-1
if(donnee(j+1)) then
constructeur_graph%matrice(i,j)=donnee(j+1)
constructeur_graph%out_count(i)=constructeur_graph%out_count(i)+1
constructeur_graph%in_count(j)=constructeur_graph%in_count(j)+1
else
constructeur_graph%matrice(i,j)=0
end if
constructeur_graph%matrice_etat(i,j)=matrice_etat(i+1,j+1)
end do
end do
do i=0,x-1
constructeur_graph%groupe(i)=tab(i+1)
end do
do i=0,x-1
k=0
do j=0,x-1
if(constructeur_graph%matrice(i,j)) then
constructeur_graph%sortant(i)%aa(k)=j
k=k+1
end if
end do
end do
do i=0,x-1
k=0
do j=0,x-1
if(constructeur_graph%matrice(i,j)) then
constructeur_graph%entrant(i)%aa(k)=j+1

```

```

k=k+1
end if
end do
end do
deallocate(donnee)
end function constructeur_graph
subroutine destructeur(grap)
type (graph) grap
integer i
if(associated(grap%matrice)) then
deallocate(grap%matrice)
end if
if(associated(grap%matrice_etat)) then
deallocate(grap%matrice_etat)
end if
if(associated(grap%in_count)) then
deallocate(grap%in_count)
end if
if(associated(grap%out_count)) then
deallocate(grap%out_count)
end if
do i=0,grap%n-1
    call destructeur_Node(grap%entrant(i))
    call destructeur_Node(grap%sortant(i))
end do
if(associated(grap%entrant)) then
deallocate(grap%entrant)
end if
if(associated(grap%sortant)) then

```

```

deallocate(grap%sortant)
end if
if(associated(grap%groupe)) then
deallocate(grap%groupe)
end if
end subroutine destructeur

integer function compteur_Noeds(grap)
type (graph) ,Pointer::grap
compteur_Noeds=grap%n
end function compteur_Noeds

logical function existe_Arc(grap,Noeud1,Noeud2)
integer ::Noeud1,Noeud2
type (graph) ,Pointer::grap
if((Noeud1>grap%n).or.(Noeud2>grap%n)) then
existe_Arc=.false.
else if (grap%matrice(Noeud1,Noeud2)) then
existe_Arc=.true.
else
existe_Arc=.false.
end if
end function existe_Arc

integer function compteur_Entrant(Noeud,grap)
integer ::Noeud
type (graph) ,Pointer::grap
compteur_Entrant=grap%in_count(Noeud)
end function compteur_Entrant

```

```
integer function compteur_Sortant(Noeud,grap)
```

```
integer ::Noeud
```

```
type (graph) ,Pointer::grap
```

```
compteur_Sortant=grap%out_count(Noeud)
```

```
end function compteur_Sortant
```

```
integer function compteur_Arc(Noeud,grap)
```

```
integer ::Noeud
```

```
type (graph), Pointer::grap
```

```
compteur_Arc=grap%out_count(Noeud)+grap%in_count(Noeud)
```

```
end function compteur_Arc
```

```
integer function GetArcEntrant(Noeud,i,grap)
```

```
integer ::Noeud
```

```
type (graph), Pointer::grap
```

```
integer ::i
```

```
GetArcEntrant=grap%entrant(Noeud)%aa(i)
```

```
end function GetArcEntrant
```

```
integer function GetArcSortant(Noeud,i,grap)
```

```
integer ::Noeud
```

```
type (graph) ,Pointer::grap
```

```
integer ::i
```

```
GetArcSortant=grap%sortant(Noeud)%aa(i)
```

```
end function GetArcSortant
```

```
end Module chain
```