

N° d'ordre : 29897

Année 1997

THÈSE
présentée
devant l'UNIVERSITÉ CLAUDE BERNARD — LYON I —
pour l'obtention
du DIPLÔME DE DOCTORAT
(arrêté du 30 mars 1992)
par Sylvie Gourlet-Fleury

MODÉLISATION INDIVIDUELLE SPATIALEMENT EXPLICITE DE
LA DYNAMIQUE D'UN PEUPEMENT DE FORÊT DENSE
TROPICALE HUMIDE
(dispositif de Paracou - Guyane française)

— ANNEXES —

ANNEXE 1

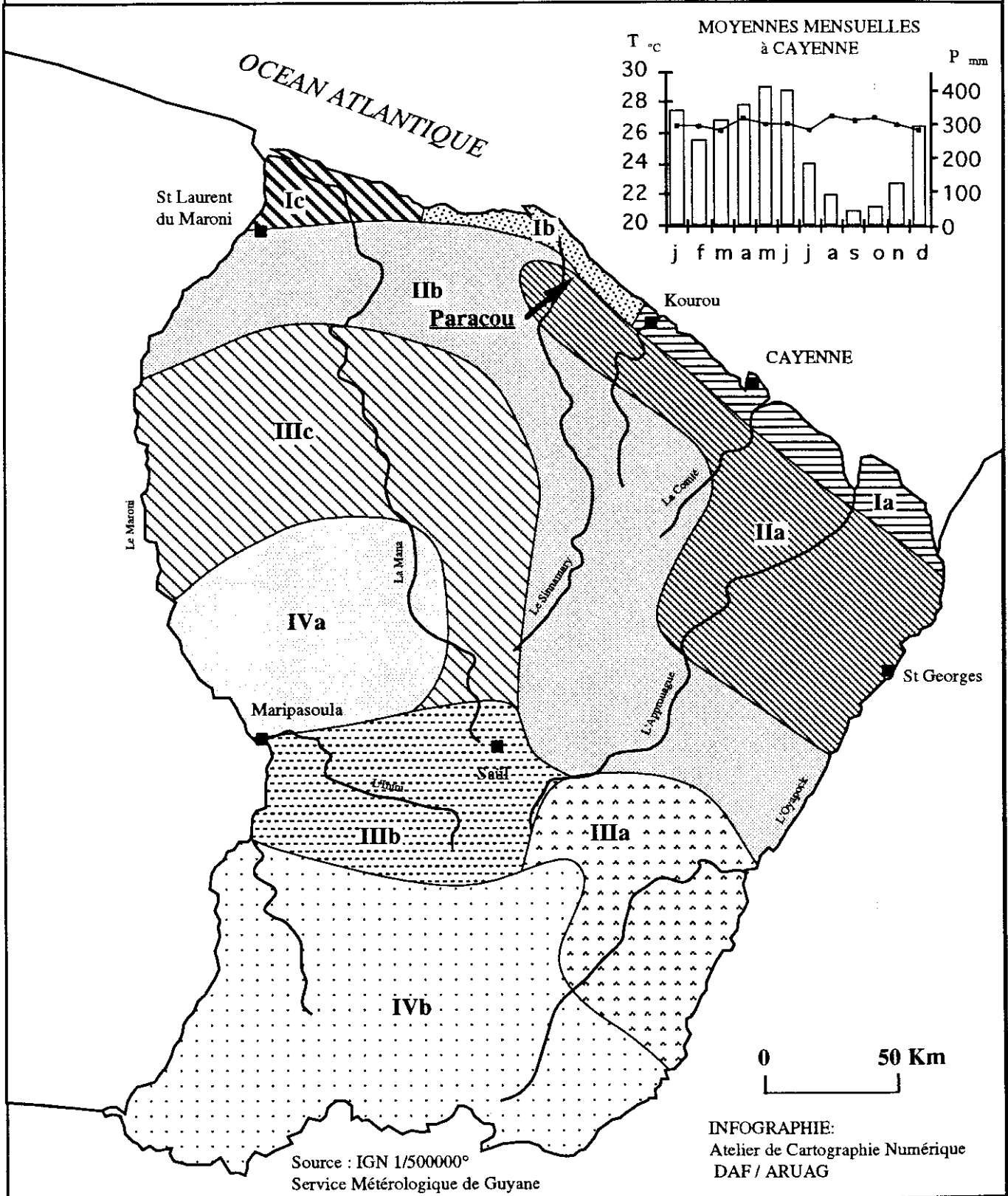
Cartes de Guyane

Carte n°1 : les grandes régions climatiques de Guyane.

Carte n°2 : grandes unités morphologiques et végétation de la Guyane.

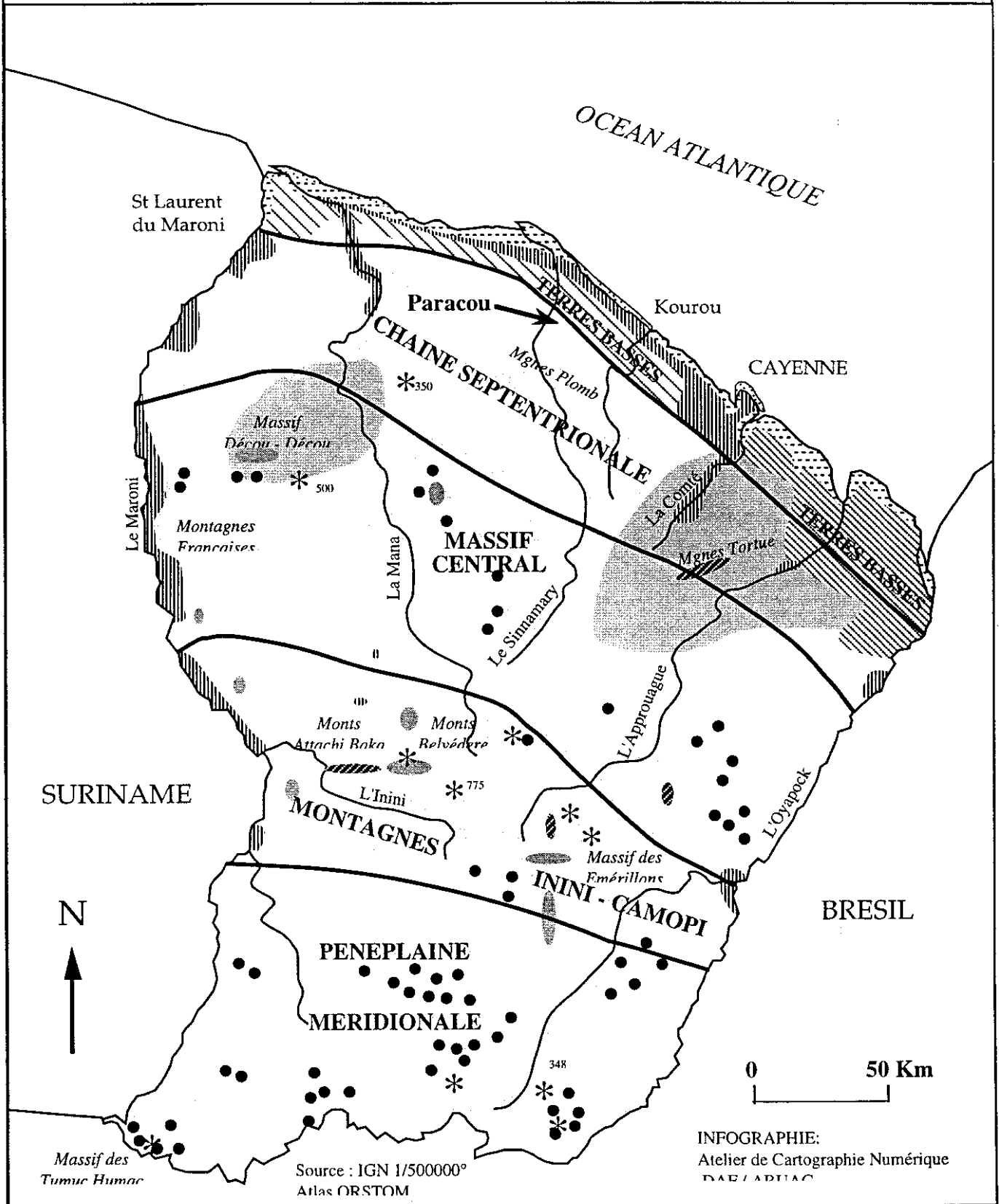
Carte n°3 : dispositif sylvicole expérimental de Paracou, recalé sur un fond de carte IGN au 1/25 000e (d'après Girou, Clément et Perthuisot, *Silvolab-Guyane*).

Carte n°1. Les grandes régions climatiques de Guyane.



- Ia** : 3 à 4 m de pluie par an (au vent de Cayenne)
- Ib** : 2 à 3 m de pluie par an (sous le vent)
- Ic** : N.O. plus sec (moins de 2 m par an)
- IIa** : 4 m et plus par an (pluviosité moyenne maximale, forte nébulosité, déficit d'insolation)
- IIb** : climat plus régulier, entre 2,5 et 3,5 m / an
- IIIa** : entre 2 et 3 m / an (influences S.E.)
- IIIb** : saison sèche peu marquée (forêt des nuages)
- IIIc** : zone de transition (faible humidité relative)
- IVa** : moins de 2 m / an, zone en "cuvette"
- IVb** : Grand Sud (moins de 1,7 m / an), zone à l'abri des dorsales topographiques NE - SW.

Carte n°2. Grandes unités morphologiques et végétation de la Guyane.

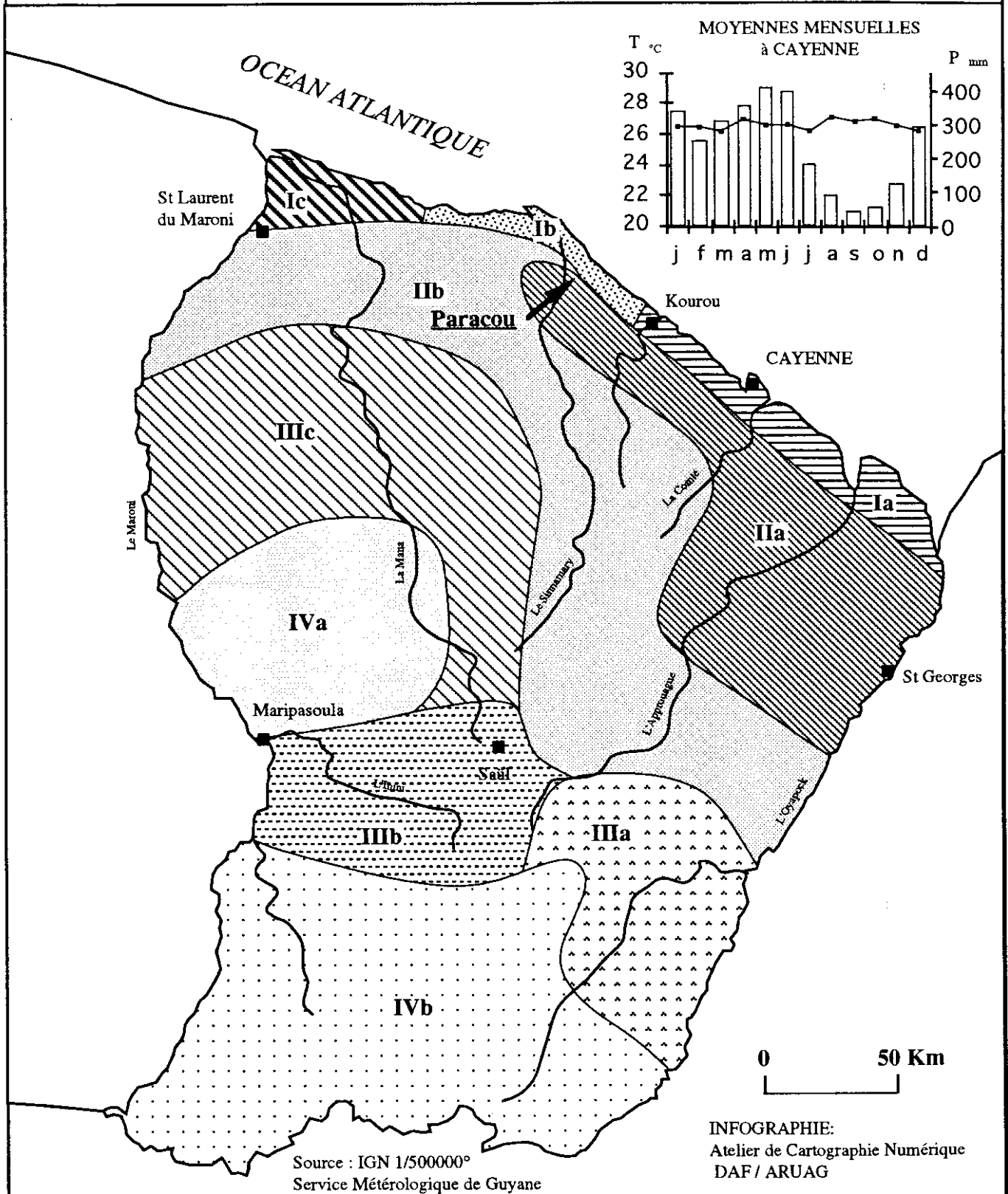


LEGENDE

- Zones de mangrove (palétuviers)
- Plages sableuses
- Formations d'origine anthropique
- Zones mixtes : savanes sèches et marécages
- Forêt inondable et marécageuse

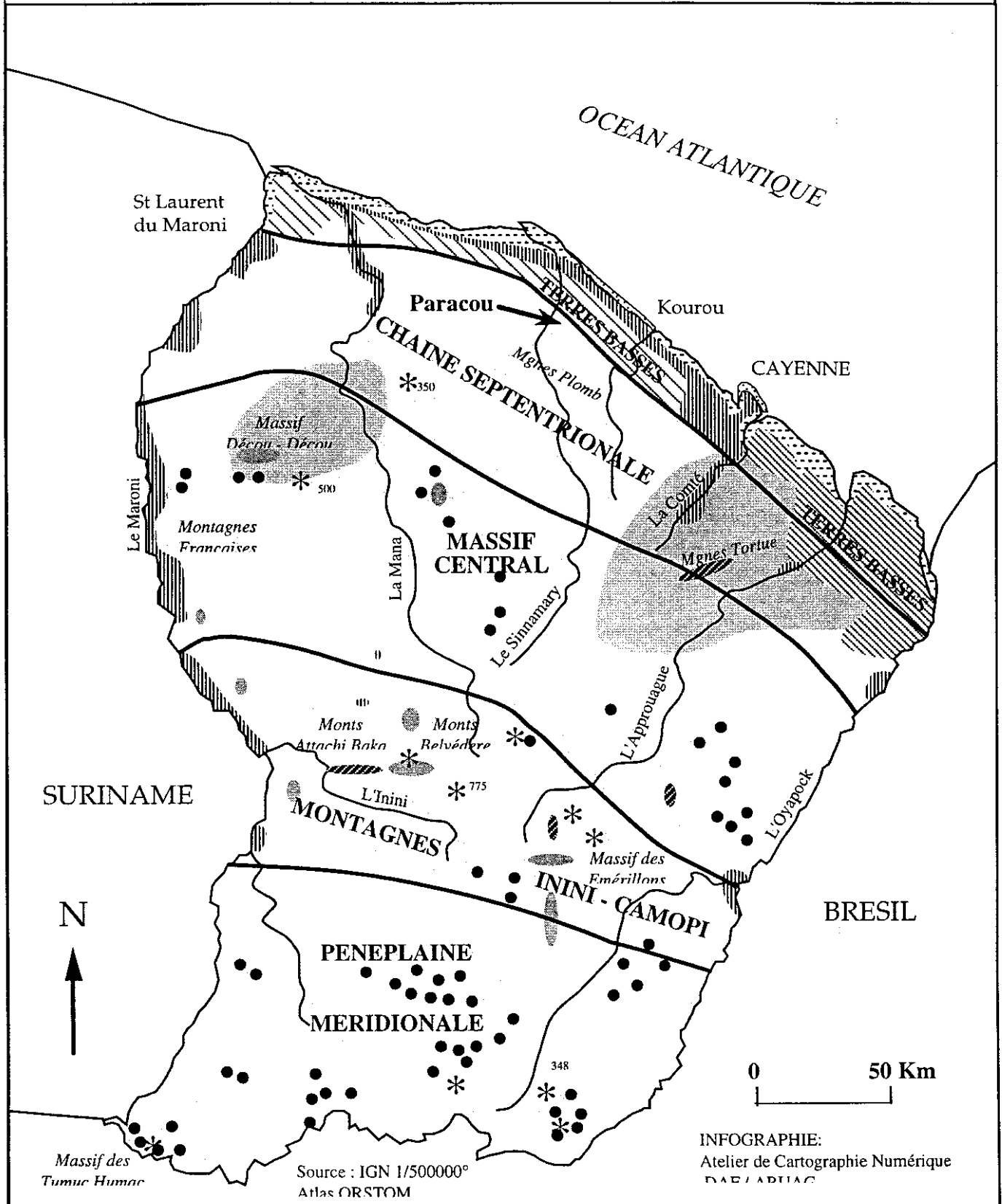
- Point côtelé
- Limite d'unité topographique
- Forêt de plaine sur sols argileux
- Forêt dense ombrophile (pluviosité)
- Forêt de "montagne", tendance orophile
- Forêt sur cuirasse latéritique
- Inselberg d'altitude (formations basses)

Carte n°1. Les grandes régions climatiques de Guyane.



- Ia** : 3 à 4 m de pluie par an (au vent de Cayenne)
- Ib** : 2 à 3 m de pluie par an (sous le vent)
- Ic** : N.O. plus sec (moins de 2 m par an)
- IIa** : 4 m et plus par an (pluviosité moyenne maximale, forte nébulosité, déficit d'insolation)
- IIb** : climat plus régulier, entre 2,5 et 3,5 m / an
- IIIa** : entre 2 et 3 m / an (influences S.E.)
- IIIb** : saison sèche peu marquée (forêt des nuages)
- IIIc** : zone de transition (faible humidité relative)
- IVa** : moins de 2 m / an, zone en "cuvette"
- IVb** : Grand Sud (moins de 1,7 m / an), zone à l'abri des dorsales topographiques NE - SW.

Carte n°2. Grandes unités morphologiques et végétation de la Guyane.

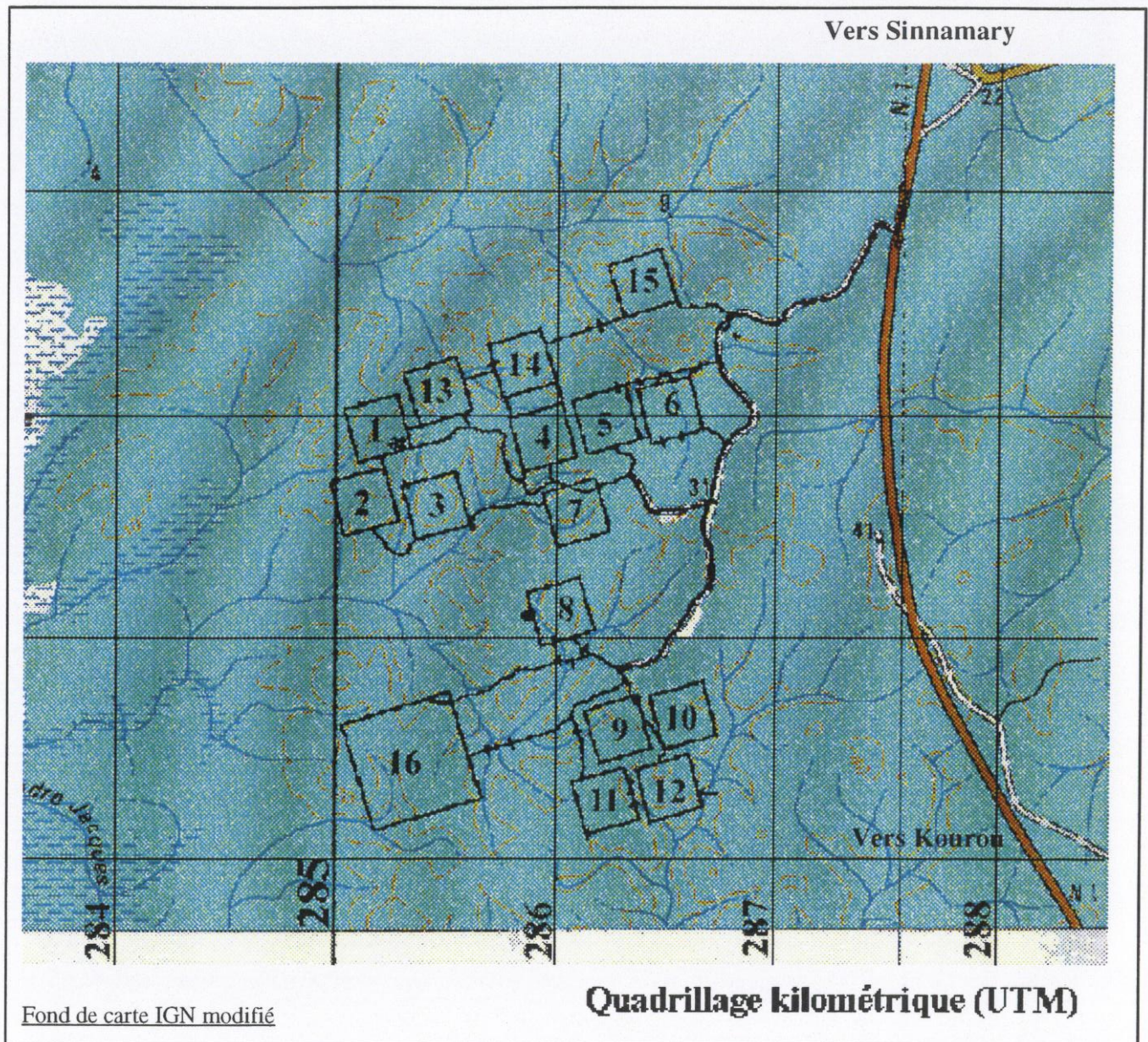


LEGENDE

- Zones de mangrove (palétuviers)
- Plages sableuses
- Formations d'origine anthropique
- Zones mixtes : savanes sèches et marécages
- Forêt inondable et marécageuse

- Point côté
- Limite d'unité topographique
- Forêt de plaine sur sols argileux
- Forêt dense ombrophile (pluviosité)
- Forêt de "montagne", tendance orophile
- Forêt sur cuirasse latéritique
- Inselberg d'altitude (formations basses)

INFOGRAPHIE:
Atelier de Cartographie Numérique
DAE/ADRIAC



Carte n°3. Dispositif sylvicole expérimental de Paracou, recalé sur un fond de carte IGN au 1/25 000e (d'après Girou, Clément et Perthuisot, *Silvolab-Guyane*).

ANNEXE 2

Quelques compléments à l'étude bibliographique

SOMMAIRE

1 - Classification des modèles.....	2
2 - Modèle matriciel de Favrichon (1995)	3
3 - Calibration des modèles de régénération de SORTIE	4
4 - Validation : aspects théoriques et pratiques	6
4.1 - <i>Qu'est-ce que la validation ?.....</i>	6
4.2 - <i>Quelques caractéristiques et statistiques utilisées pour l'évaluation quantitative des modèles.....</i>	7

NB. Les références bibliographiques renvoient à la bibliographie générale

1 - Classification des modèles

Legay (1973) oppose les modèles *physiques* aux modèles *dialectiques* et, parmi ces derniers, les modèles *discursifs* (c'est-à-dire ceux du discours parlé, puis écrit) aux modèles *formels*, nés des précédents par un appauvrissement de l'expression et l'usage de symboles. Les modèles *formels* sont classés en (i) modèles *analytiques* ou *déterministes*, qui rendent compte directement de phénomènes supposés déterminés par un petit nombre de causes parfaitement connues ; ils ne comportent aucune composante aléatoire (Law et Kelton, 1991) et (ii) modèles *probabilistes* (ou *stochastiques*, selon les auteurs) qui rendent seulement compte de la probabilité de phénomènes supposés déterminés par des causes qui ne sont pas toutes contrôlées, et peuvent être en très grand nombre. Ces modèles fournissent des résultats aléatoires, qui doivent être traités comme de simples estimations des propriétés réelles des modèles (Law et Kelton, 1991). Pavé (1994) fait remarquer qu'il existe des ponts entre les approches stochastique et déterministe, qui s'intéressent simplement à des niveaux d'organisation différents. Le modèle déterministe s'intéresse à un niveau « supérieur » d'organisation et peut être déduit du modèle stochastique par « passage à la moyenne ». En ce sens, on parle également d'approche *mécaniste* (probabiliste, stochastique) et d'approche *phénoménologique* (déterministe), cette dernière étant assimilée par Grimm (1994) à une approche *conceptuelle*, ou encore *stratégique*. Un parallèle peut ici être établi avec l'opposition entre modèles *fonctionnels* et modèles *empiriques* en matière de modélisation de la croissance des arbres ou des peuplements (Turnbull, 1963, Korzhukhin et al., 1996) ; selon Houllier (1992) :

- à une échelle donnée, un modèle *fonctionnel* est un modèle qui incorpore des connaissances sur les mécanismes biologiques, connaissances qui ressortent généralement de niveaux de description plus fins. On parle aussi de « *biologically constrained model* », de « *process-based* » ou « *process model* », de « *mechanistic model* » (Ek et Dudek, 1980, Korzhukhin et al., 1996), de modèle « *théorique* » (Mortier, 1990) ou de « *modèle physiologique* », Deleuze (1992), Garcia (1994). Bruce et Wensel (1988) donnent des « *process models* » une définition assez restrictive : « *Process models simulate the biological processes that convert carbon dioxide, nutrients, and moisture into biomass through photosynthesis.* ». Mohren et Burkhardt (1994) précisent que ces modèles sont alors des « *biologically-based process models* ». D'une manière générale, ces termes prêtent à confusion et sont utilisés différemment selon les auteurs. Remarque : il est bien entendu que des modèles à base physiologique peuvent être élaborés à toutes les échelles, et pas seulement celle de l'arbre ;
- à cette même échelle, un modèle est *empirique* (ou *dendrométrique*, Deleuze, 1992, ou encore *descriptif*, Pavé, 1994) quand il vise à rendre compte des phénomènes observés à partir d'expérimentations, de placettes permanentes ou de données d'inventaire, mais qu'il n'intègre pas explicitement de connaissances sur les processus élémentaires de croissance qui génèrent ces phénomènes.

Korzhukhin et al. (1996) donnent une série d'exemple de modèles empiriques et de modèles fonctionnels développés depuis une vingtaine d'années aux différentes échelles pouvant intéresser les forestiers : échelle de l'arbre, du peuplement, du paysage. Ils soulignent que la limite entre fonctionnel et empirique peut être très floue et que la plupart des modèles typiquement « *process-based* » (*ie* à base physiologique) comportent des relations empiriques. Reynolds et Acock (1985) analysant les modèles de croissance utilisés pour tester les effets sur la végétation d'une augmentation du CO₂ dans l'atmosphère font le même constat. Il est toutefois intéressant de noter qu'ils classent les « *gap models* » (modèles de trouées, voir §

2.2.3.1 du texte principal) dans la catégorie des modèles empiriques, alors que Korzukhin et al. (1996) considèrent qu'il s'agit de modèles fonctionnels (nous les avons classés dans cette dernière catégorie).

2 - Modèle matriciel de Favrichon (1995)

Auteur du premier modèle de dynamique forestière appliqué aux peuplement du dispositif de Paracou, Favrichon utilise une matrice de Usher généralisée permettant de s'affranchir partiellement de l'hypothèse de stationnarité. Il fait pour cela intervenir une régulation des paramètres par des variables décrivant l'état du peuplement considéré sur un carré de 1,56 ha (cf. § 1.2.2.1 du texte principal). Le pas utilisé dans les simulations est de 2 ans et le modèle général est le suivant :

$$N_{t+1} = P_t(I - M_t)N_t + R_t$$

avec :

- N_t et N_{t+1} vecteurs d'état du peuplement respectivement aux temps t et $t+1$. Il s'agit des effectifs par classes de diamètre de 5 cm ;
- P_t = matrice de passage, dont les éléments dépendent du diamètre médian de chaque classe et du rapport $\frac{G_t}{G_0}$ des surfaces terrières totales du peuplement respectivement aux temps t et t_0 ; soient i la $i^{\text{ème}}$ classe de diamètre, D_i le diamètre médian correspondant :

$$p_{i,i+1} = a_0 + a_1 D_i + a_2 D_i^2 + a_3 D_i^3 + b_0 + b_1 \frac{G_t}{G_0}.$$

a_0 , a_1 , a_2 et a_3 sont des coefficients ajustés sur les parcelles témoins et b_0 et b_1 sont ajustés sur les parcelles traitées, sachant les précédents ;

- M_t = vecteur décrivant la mortalité comme somme d'une mortalité naturelle (les coefficients sont considérés constants et égaux à 2% sur 2 ans, la dernière classe de diamètre se voyant affecter un taux de 5%) et le cas échéant, d'une mortalité due à une intervention sylvicole puis en cas d'exploitation forestière, induite par l'intervention. Cette dernière est déduite du nombre d'arbres exploités NXT : $m_i = c_0 + c_1 NXT$ avec c_0 et c_1 coefficients estimés ;
- R_t = vecteur de recrutement. Seul le premier terme est différent de 0 car on néglige le recrutement direct dans les classes de diamètres autres que la première. Il est régulé par l'effectif total présent sur le carré de 1,56 ha :

$$r_{1,t} = d_0 e^{(-d_1 \frac{N_t}{N_{t0}})} \text{ avec } d_0 \text{ et } d_1 \text{ coefficients estimés.}$$

Les régulations introduites permettent de tenir compte de l'effet induit par différents types d'intervention sylvicole sur les principales composantes de la dynamique forestière. Le réalisme du modèle est ensuite renforcé par la prise en compte de 5 groupes d'espèces présentant des comportements « dynamiques » plus ou moins homogènes, ce qui permet à l'auteur d'évaluer l'influence de différents scénarios sylvicoles sur l'évolution de la composition floristique des peuplements.

3 - Calibration des modèles de régénération de SORTIE

La méthode utilisée pour décrire les processus de régénération dans SORTIE (Pacala et *al.*, 1993, Pacala et *al.*, 1996) est exposée dans Ribbens et *al.* (1994). Le principe consiste à déterminer, pour chaque espèce présente dans le peuplement (10 espèces dans le cas de SORTIE) la quantité moyenne de plantules viables produites par un arbre « adulte » moyen ainsi que leur fonction de répartition dans l'espace, toujours à partir de l'adulte moyen. Pour cela, on procède selon la méthode décrite ci-dessous.

- 1) Des transects constitués de quadrats contigus de 1 m² sont matérialisés dans le peuplement selon des directions aléatoires. Toutes les plantules jugées viables (en pratique, il s'agit des plantules ayant germé pendant l'année ou bien, pour certaines espèces moins bien représentées, des individus faisant moins de 25 cm de hauteur) sont recensées à l'intérieur des quadrats et tous les reproducteurs potentiels, à savoir les individus de plus de 10 cm de diamètre, sont cartographiés dans un rayon de 20 m autour de chaque transect. Lorsque les espèces concernées sont dioïques, seuls les individus femelles sont pris en compte ;
- 2) On suppose que le nombre de plantules installées dans chaque quadrat suit une loi de Poisson, dont la moyenne dépend de la taille du pied-mère potentiel et de la distance à laquelle il est situé. La fonction permettant de calculer cette moyenne est constituée de deux composantes :

$C_1 = STR \left(\frac{D}{30} \right)^2$, avec D = diamètre du reproducteur potentiel et STR = « standard total recruitment », nombre de plantules viables installées par arbre « standard », ici un « adulte » de 30 cm de diamètre.

$C_2 = \frac{1}{n} e^{-DFm^3}$ et $n = \int_0^\infty 2\pi m \left[e^{-DFm^3} \right] dm$, avec m = distance entre le centre du quadrat et le reproducteur potentiel et DF = facteur de dispersion, déterminant la rapidité du déclin de la densité de plantules en fonction de la distance au pied-mère. Il permet d'évaluer une distance moyenne de dispersion. C₁ représente le nombre total de plantules viables installées par reproducteur et C₂ la densité moyenne de plantules viables installées sur un quadrat de 1 m² situé à une distance m du reproducteur, sous l'hypothèse que la répartition autour de celui-ci est isotrope. Le nombre P de plantules installées par un pied-mère donné sur un quadrat donné se calcule donc comme : $P = C_1 * C_2$. Sachant que sur chaque quadrat tous les pieds-mères potentiels recensés lors de l'inventaire ont pu installer des plantules viables, on déduit des équations précédentes que le nombre total de plantules d'une espèce donnée présentes sur un quadrat i étant donné T adultes repérés est :

$$R_i = \sum_{j=1}^T (STR) \left(\frac{D_j}{30} \right)^2 \frac{1}{n} e^{-DFm_{ij}^3} \quad (m_{ij} : \text{distance entre le } i^{\text{ème}} \text{ quadrat et le } j^{\text{ème}} \text{ adulte}).$$

- 3) L'estimation des paramètres STR et DF est réalisée par maximisation de la fonction de vraisemblance décrivant la probabilité d'observer les effectifs O_i recensés par inventaire sur les différents quadrats, lorsqu'une moyenne R_i est attendue sous les hypothèses précédemment énoncées.

Les résultats obtenus par Ribbens et *al.* (1994) montrent notamment que :

- les courbes de dispersion de plantules obtenues pour les différentes espèces sont cohérentes avec leur mode de dissémination respectif ; en particulier, la distance moyenne de dispersion chez les espèces anémochores est plus importante que chez les espèces

zoochores. L'inventaire des adultes dans un rayon de 20 m autour des transects s'est révélé tout-à-fait suffisant pour 9 espèces sur 10, la dernière étant une anémochore à grand rayon de dispersion pour laquelle le paramètre DF est mal estimé ;

- l'estimation du paramètre DF s'est révélée être très robuste lors d'opérations de validation diverses, sur le même site, sur des sites différents et sur des périodes de travail différentes ;
- l'estimation du paramètre STR pose davantage de problèmes, du fait de l'existence d'une très grande variabilité spatio-temporelle dans la phénologie des espèces.

La calibration de tels modèles, permettant de quantifier sur des bases objectives et à l'aide d'un inventaire simple deux informations aussi complexes à obtenir en forêt hétérogène que la production moyenne d'un reproducteur et la distance moyenne de dispersion, présente différents avantages :

- selon l'objectif assigné au modèle de dynamique, selon les mécanismes que l'on cherche à étudier, le lien entre pieds-mères et descendance peut être établi à différents stades de la régénération selon la même méthode moyennant éventuellement l'ajustement du protocole : un couplage avec un modèle de dynamique d'animaux prédateurs de graines, par exemple, impliquerait de s'intéresser davantage aux graines parvenues dans chaque quadrat qu'aux plantules. Si l'objectif consiste à tester l'effet de traitements sylvicoles divers sur la production à moyen terme d'espèces de valeur et si l'incertitude pesant sur le devenir des plantules de l'année est trop importante, on s'intéressera davantage à des individus plus âgés (critère de taille quelconque). On conçoit cependant qu'il ne soit pas possible de dépasser une certaine limite au-delà de laquelle la faible abondance de juvéniles ne permettra plus d'estimer correctement les paramètres STR et DF, et où le lien sera perdu ;
- elle fournit une méthode simple de comparaison entre espèces et une méthode potentielle de classification en « groupes de régénération » fort utile dans les milieux très riches en espèces, lorsque la phénologie et les modes de dissémination ne sont qu'imparfaitement connus et peu ou pas quantifiés ;
- l'implémentation dans un simulateur spatialement explicite est relativement aisée à réaliser. Ek et Monserud (1974a, Monserud, 1975) ont construit leur modèle de régénération à partir de paramètres (STR et DF) de même nature, mais en utilisant des données de la littérature, sans possibilité de réaliser de validation autre que qualitative et portant sur le fonctionnement global du modèle.

La mise en oeuvre d'une telle méthode à Paracou, moyennant quelques adaptations, nous paraît prometteuse au moins pour les espèces présentant des modes de dissémination de type anémochore à courte distance, barochore, autochore et synrodontochore (dissémination par des rongeurs comme l'agouti — *Dasyprocta leporina* — ou l'acouchi — *Myoprocta exilis* —, voir Forget, 1988, Collinet, 1997) dont les distances moyennes de dispersion ont toutes chances d'être compatibles avec un protocole raisonnable. Cela concerne des espèces intéressantes à divers titres, notamment des espèces exploitées dont on ignore largement quelles seront les conséquences de l'exploitation des adultes sur la dynamique de populations à répartition spatiale bien typée (*Dicorynia guianensis*, *Qualea rosea*, *Eperua falcata*, *Vouacapoua americana*, etc.) : or, il s'agit typiquement là d'une question dont l'exploration peut être envisagée à l'aide de modèles individuels spatialisés.

4 - Validation : aspects théoriques et pratiques

4.1 - Qu'est-ce que la validation ?

« If a model is not *valid*, then any conclusions derived from the model will be of doubtful value » (Law et Kelton, 1991). Tout le problème consiste à savoir quel contenu on donne au mot « validation ». Selon Oreskes et *al.* (1994) ce terme, à connotation positive, peut revêtir un sens différent et souvent abusif selon les auteurs qui l'utilisent :

- « In some cases, validation is used interchangeably with verification to indicate that model predictions are consistent with observational data. Thus, modelers misleadingly imply that validation and verification are synonymous, and that validation establishes the veracity of the model. » (Siekierski, 1991, par exemple) ;
- « In other cases, the term validation is used even more misleadingly to suggest that the model is an accurate representation of physical reality ». C'est le sens de la définition qu'en donnent, par exemple, Law et Kelton (1991 : valider un modèle consiste à déterminer s'il représente de manière précise — « accurate » — le système réel étudié) ou Alder (1995).

Oreskes et *al.* (1994) soulignent que le terme de « verification » implique l'affirmation ou l'établissement d'une vérité et qu'il n'est pas possible de démontrer qu'une proposition est vraie, sauf dans un système fermé. Or les modèles utilisés en écologie, de par les paramètres mal connus, les données incomplètes, les inférences, les hypothèses simplificatrices sous-jacentes qu'ils utilisent, ne sont jamais des systèmes fermés. « In its application to models of natural systems, the term verification is highly misleading. It suggests a demonstration of proof that is simply not accessible ». Ils ajoutent que : « What typically passes for validation and verification is at best confirmation, with all the limitations that this term suggests. », la « confirmation » consistant à établir que des observations empiriques ne contredisent pas le modèle ou l'hypothèse explicatifs sous-jacents. Caswell (1976), Oderwald et Hans (1993) utilisent le terme de « corroboration » dans ce même sens : « the goal is to attempt to refute the model, and our confidence in its statements increases as it passes more and more severe tests. » (Caswell, 1976).

Il semble donc qu'il vaille mieux parler, d'une manière générale, d'évaluation des performances d'un modèle au regard d'une réalité observée, plutôt que de « validation ». Cette évaluation des performances peut revêtir différents aspects qualitatifs et quantitatifs (voir notamment Oderwald et Hans, 1993, Bart, 1995, Soares et *al.*, 1995, Pacala et *al.*, 1996) et doit intervenir, à des titres divers, tout au long du travail de modélisation :

- 1) examen du modèle et de ses composantes en termes de logique interne et de réalisme biologique (ce que Oderwald et Hans, 1993, qualifient de « *corroboration* », Vanclay, 1994, de « *model criticism* » et Power, 1993, de « *structural validation* »), vérification au sens propre du terme de la logique des signes dans les équations ou bien du code informatique (Law et Kelton, 1991), etc. ;
- 2) caractérisation des erreurs au travers de l'étude des écarts entre valeurs observées et prédites et tests statistiques éventuels, soit sur le jeu de données qui a servi à estimer les paramètres des modèles, soit sur un jeu de données indépendant : Power (1993) parle respectivement de « *replicative validation* » et de « *predictive validation* », Alder (1995) de « *self-validation* » et de « *independent validation* ». Vanclay (1994) qualifie plus généralement ce travail de « *benchmarking* » ;

- 3) étude de la propagation des erreurs dans le modèle global par exploration de l'intervalle de confiance des paramètres, étude de la sensibilité du modèle à différentes valeurs de certains paramètres auxiliaires non estimés, étude de la plausibilité du comportement qualitatif, étude des réponses fournies par le modèle à un signal donné (voir Luckyanov, 1993, Pacala et *al.*, 1996).

Legay (1973) et Soares et *al.* (1995) insistent sur le fait que l'évaluation d'un modèle doit autant que possible reposer sur un ensemble complet de critères, de manière à se faire une idée d'un véritable « profil » comportemental. Selon l'objectif du modèle construit, il est clair cependant que certains aspects sont davantage privilégiés que d'autres (Caswell, 1976, Oderwald et Hans, 1993, Garcia, 1994, Soares et *al.*, 1995 : « evaluation is problem-dependent ») : le point 2) est essentiel pour les modèles de prévision, les points 1) et 3) paraissent les plus importants dans le cas des modèles de mécanismes, un travail approfondi sur le point 1) peut suffire pour un modèle pédagogique, etc. L'importance d'une définition claire des objectifs poursuivis par toute entreprise de modélisation s'en trouve renforcée : « Failure to provide detailed descriptions of model objectives has been a frequent problem. (...) the great majority of criticisms of models relate to a capacity for which the model was not designed in the first place. » (Bart, 1995, citant Overton, 1977).

Quelle confiance accorder à l'évaluation des performances d'un modèle ? Oreskes et *al.* (1994) insistent sur un certain nombre d'éléments à conserver en mémoire :

- « More than one model construction can produce the same output » ;
- « if a model fails to reproduce observed data, then we know that the model is faulty in some way, but the reverse is never the case ». Ce point est souligné par différents auteurs (Caswell, 1976, Oderwald et Hans, 1993) qui en concluent que les échecs de validation sont plus intéressants et riches en enseignements que les succès qui, de toute façon, ne prouvent rien ;
- « even if a model result is consistent with present and past observational data, there is no guarantee that the model will perform at an equal level when used to predict the future. ». Les systèmes naturels, notamment, sont dynamiques et peuvent évoluer de manière inattendue ;
- « No matter how much data we have, there will always be the possibility that more than one theory can explain the available observations ».

Ces problèmes revêtent une importance critique lorsque l'on travaille sur des modèles de prévision dont dépendent des choix touchant à la santé et à la sécurité publiques, ou aux investissements économiques et militaires. Dans le domaine qui nous occupe, ils doivent inciter à la prudence dans les conclusions et à la modestie dans les prétentions ...

4.2 - Quelques caractéristiques et statistiques utilisées pour l'évaluation quantitative des modèles

Dans la plupart des articles traitant de l'évaluation quantitative des modèles, les auteurs se placent dans le cas où les données observées n'ont pas été utilisées durant la phase de construction du modèle (nous parlerons de « jeu de données indépendant »). Cela n'empêche pas que les calculs effectués dans ce cadre puissent également l'être en utilisant les données de calibration pour caractériser la qualité des régressions effectuées.

Soient O_i et P_i des valeurs quantitatives respectivement observée et prédite par un modèle quelconque sur un individu i (ce peut être un arbre ou un ensemble d'arbres dans le cas qui nous intéresse) appartenant à un jeu de données indépendant. Après examen graphique de la distribution des écarts entre valeurs, des liaisons éventuelles subsistant avec les diverses variables prédictives, etc., on peut calculer, pour les n individus du jeu :

- le biais moyen : $B = \frac{\sum_{i=1}^n (O_i - P_i)}{n}$ (nul, si le jeu de données a été utilisé pour l'ajustement d'un modèle linéaire). Sous l'hypothèse $H_0 : B = 0$, la variable $T = \frac{\sqrt{n} * B}{s}$ (avec s écart-type résiduel estimé sur l'échantillon ayant servi à ajuster le modèle) suit approximativement une loi normale centrée réduite (Power, 1993, d'après Hogg and Tanis, 1988 et Abraham et Ledolter, 1983). Freese (1960) souligne le danger de ce test, qui peut conduire à rejeter l'hypothèse H_0 pour des biais de faible importance lorsque s est faible, *ie* le modèle est précis ou à ne pas le rejeter lorsque le modèle est peu précis et le biais important. Il propose d'utiliser une statistique équivalente, à savoir le rapport $\frac{\sum (O_i - P_i)^2}{s_u^2}$ qui, sous l'hypothèse H_0 , suit un χ^2 à $(n-2)$ degrés de liberté ; s_u^2 n'est plus la variance résiduelle estimée, mais définie a priori par l'utilisateur comme le degré de précision qu'il attend des prédictions de son modèle. Reynolds (1984) indique comment construire des intervalles de confiance, de prédiction et de tolérance sur le biais, de manière à estimer correctement l'amplitude de l'erreur commise en utilisant le modèle pour prédire O_i ;

- l'estimation de la variance résiduelle (ou écart-quadrique moyen) : $MSE (= \bar{\sigma}^2) = \frac{\sum_{i=1}^n (O_i - P_i)^2}{n-1}$ et sa racine carrée (RMSE).

Il convient de noter que, lorsque le biais est significativement non nul, l'estimation non biaisée de la variance résiduelle est :

$$\bar{\sigma}^2 = \frac{\sum_{i=1}^n (O_i - P_i - B)^2}{n-1}.$$

Mayer et Butler (1993) signalent d'autres mesures des écarts, moins fréquemment utilisées : « mean absolute error » (MAE) et « mean absolute percent error » (MA%E) définies comme suit :

$$MAE = \frac{\sum |O_i - P_i|}{n} \text{ (même unité que les données étudiées) et}$$

$$MA\%E = 100 * \frac{\sum |O_i - P_i| / |O_i|}{n} \text{ (sans unité).}$$

Kleijnen (1987) cité par Mayer et Butler (1993) recommande un seuil de $MA\%E = 10\%$ comme limite « d'acceptabilité » pour un modèle en cours de validation, mais les auteurs conviennent qu'il n'est pas réaliste de vouloir fixer une telle limite et qu'en réalité, ces mesures sont utiles essentiellement pour comparer entre elles les performances de différents modèles. Il faut noter que $MA\%E$ est très sensible aux faibles valeurs de O_i et que de ce fait, des mesures telles que $\frac{MAE}{O}$ ou $\frac{RMSE}{O}$ peuvent la remplacer avec avantage ;

- la statistique Q (Power, 1993, d'après Abraham et Ledolter, 1983) sur la somme des carrés

des erreurs : $Q = \frac{\sum_{i=1}^n (O_i - P_i)^2}{n \cdot s^2}$, (toujours avec s écart-type résiduel estimé sur l'échantillon

ayant servi à calibrer le modèle). Sous l'hypothèse H_0 d'égalité des variances résiduelles estimées respectivement sur le jeu de données de calibration et le jeu indépendant, cette statistique suit un F à $(n, m-p)$ degrés de liberté (avec p : nombre de paramètres du modèle ajusté et m : nombre d'individus ayant servi à ajuster le modèle). A noter que cette statistique est très sensible aux points aberrants ;

- le pseudo- R^2 (Chai et Lemay, 1993) ou « modelling efficiency » (Mayer et Butler, 1993, d'après Loague et Green, 1991, Soares et *al.*, 1995, Ong et Kleine, 1996), sans dimension, qui donne une indication globale de l'ajustement du modèle aux données :

$$\text{pseudo-}R^2 = EF = 1 - \frac{\sum (O_i - P_i)^2}{\sum (O_i - \bar{O})^2}.$$

Il s'agit d'une extension de la notion de coefficient de détermination (R^2) à la classe des modèles de régression non linéaire. Alors que $R^2 \in [0, 1]$, le pseudo- $R^2 \in [-\infty, 1]$, les valeurs négatives indiquant un sérieux problème d'adéquation aux données ;

- la statistique D de Kolmogorov pour tester la normalité de la distribution des résidus : $D_{\text{obs}} = \text{Sup} |F_e(x) - F_N(x)|$ est comparée à la valeur critique $D_\alpha = C_\alpha \cdot \sqrt{2/n}$, avec F_e et F_N respectivement fonctions de distribution des erreurs et de la loi normale, et C_α fournie par un tableau en fonction de α (Tomassone et *al.*, 1993, par ex.). Plus généralement, cette statistique peut être utilisée pour comparer deux distributions, comme le fait Doyle (1981, cité par Shugart, 1984) pour tester les prédictions de son modèle FORICO ;
- le test F simultané sur la « régression des données observées sur les données prédites ». On effectue la régression $O_i = aP_i + b$ et on calcule la statistique suivante :

$$F = \frac{nb^2 + 2b(a-1)\sum P_i + (a-1)^2\sum P_i^2}{2s^2} \quad (s = \text{écart-type résiduel de la régression})$$

qui, sous l'hypothèse H_0 : $a=1$ et $b=0$ suit un F à $(2, n-2)$ degrés de liberté (Mayer et *al.*, 1994, d'après Dent et Blackie, 1979). Power (1993) et Soares et *al.* (1995) indiquent l'intérêt de ce test. Mayer et *al.* (1994), utilisant des simulations de Monte-Carlo, soulignent également son intérêt par rapport à d'autres tests sur les données appariées tels que le test- t (Freese, 1960) ou le test des rangs signés de Wilcoxon et recommandent son usage en soulignant qu'il est très performant dans le rejet de modèles invalides. Cependant, ces auteurs basent leurs expériences sur des données longitudinales (auto-corrélées dans le temps), et Thornton et Hansen (1996) réfutent leurs conclusions dans le cas où les données ne présentent pas ces caractéristiques. D'après leurs propres simulations, le test F simultané entraînerait un fort taux de rejet erroné de l'hypothèse H_0 , d'autant plus fort que la taille de l'échantillon est importante et le coefficient de corrélation $r(O, P)$ faible. Nous n'utilisons ce test qu'à titre indicatif.

Parmi les méthodes proposées par Smith et Rose (1995) pour évaluer la qualité d'un modèle, la régression $O_i = aP_i + b$ tient une place importante, mais ils n'utilisent pas le test F simultané. Il leur paraît préférable de tester séparément les deux hypothèses H_0 : $a=1$, « which tests that the observed values and predicted values vary together consistently over their ranges » puis H_0 : $b=0$ qui indique l'absence de biais dans les prédictions (si

l'hypothèse $a=1$ n'a pas été rejetée). La somme des carrés résiduels $\sum (O_i - P_i)^2$ peut ainsi être décomposée en faisant apparaître un terme de biais, un terme traduisant le degré de « consistance » du modèle et un terme d'erreur résiduelle liée à la régression linéaire $O_i = aP_i + b$ (Theil, 1958, cité par Smith et Rose, 1995). Les deux tests séparés sont également retenus par Pavé (1994) et Mesplé *et al.* (1996). A noter cependant que le test F-simultané peut conduire à rejeter l'hypothèse nulle quand les deux tests séparés ne le font pas : des exemples démonstratifs figurent dans Mayer et Butler (1993) et dans Soares *et al.* (1995).

- La mesure d'association U , proposée par Theil (1958, cité par Smith et Rose, 1995) :

$$U = \frac{\sqrt{\frac{1}{n} \sum (O_i - P_i)^2}}{\sqrt{\frac{1}{n} \sum O_i^2} + \sqrt{\frac{1}{n} \sum P_i^2}}$$

U mesure une distance, et non une similarité comme le coefficient de corrélation. Elle varie entre 0 (adéquation parfaite des prédictions aux données observées) et 1 (obtenu lorsque toutes les valeurs observées, ou toutes les valeurs prédites valent 0). Des valeurs supérieures à 0,5 indiquent un gros problème dans la formulation du modèle.

- Reynolds *et al.* (1981) proposent toute une série de tests paramétriques et non paramétriques plus ou moins complexes de validation statistique d'un modèle, en se plaçant dans le cas où les prédictions sont issues de simulations répétées : on dispose alors de n -uplets $(O_i, P_{1i}, P_{2i}, \dots, P_{ni})$ et non plus de simples couples (O_i, P_i) . Ils sont critiqués par Mayer et Butler (1993) qui démontrent que des tests simples classiques (listés plus haut) sont tout aussi performants sur les exemples fournis dans leur article. En fait, Reynolds *et al.* indiquent que les tests qu'ils présentent sont davantage destinés à mettre en évidence des « subtiles discrepancies between the model and the real system ».
- Power (1993) propose également des statistiques destinées à comparer les prédictions fournies par différents modèles. Certaines sont classiques (biais moyen, MSE, MAE, MA%E : cf. plus haut), d'autres moins comme le « coefficient d'inégalité de Theil » ou le « coefficient de Janus » (resp. Theil, 1966 et Gadd et Wold, 1964, cités par Power, 1993). Power se place dans le cadre de données longitudinales et d'une « cross-validation » faite en fonction du temps : le modèle est ajusté sur les données de la période $(0, t)$ et la validation réalisée sur les données de la période $(t+1, T)$. Le coefficient de Janus nous a paru intéressant à calculer également pour des données transversales. Si le jeu de données de calibration contient m valeurs, son expression est la suivante :

$$J^2 = \frac{\sum_{i=1}^n \frac{(O_i - P_i)^2}{n}}{\sum_{j=1}^m \frac{(O_j - P_j)^2}{m}}$$

avec i et j se rapportant respectivement aux données indépendantes et de calibration. « The Janus quotient thus forms the ratio of the predictive and replicative validity measures to provide an index of the change in model accuracy between the estimation and prediction data sets » (Power, 1993). Une valeur élevée de J^2 peut indiquer par exemple un changement de structure dans les données, ou bien qu'une importance trop grande a été

accordée à des « bruits de fond » présents dans le jeu d'ajustement, résultant en une surparamétrisation du modèle.

Peu d'auteurs parlent de l'évaluation quantitative des modèles de régression logistique. Différents tests sont signalés dans Norusis (1990) et Trexler et Travis (1993), dont :

- le test de la déviance : $-2\text{Log}(\text{vraisemblance})$ possède une distribution du χ^2 à $(n-p)$ degrés de liberté, si p est le nombre de paramètres du modèle logistique. Cela permet de tester l'hypothèse H_0 : le modèle ajusté ne diffère pas d'un modèle parfait pour lequel la vraisemblance vaudrait 1 ;
- la statistique de « goodness-of-fit » : $Z^2 = \sum \frac{(O_i - P_i)^2}{P_i(1 - P_i)}$, qui suit également un χ^2 à $(n-p)$ degrés de liberté et permet de tester la même hypothèse.

Remarque : dans l'absolu et même si l'inadéquation du modèle aux données qu'il est censé prédire est aisément mise en évidence, le calcul de ces différentes statistiques peut tout simplement permettre d'évaluer l'intensité des problèmes rencontrés.

ANNEXE 3

Modèle de Korf et modèle de Gompertz

Modèle de Korf (également connu sous le nom de modèle de « Lundqvist-Matern »)

- Equation analytique : $y(t) = Ke^{\left(\frac{-1}{at}\right)^m}$.
- Equation différentielle : $\frac{dy}{dt} = amy[\text{Log}(K) - \text{Log}(y)]^{\left(1+\frac{1}{m}\right)}$.
- Position du point d'inflexion : $y = Ke^{-\left(1+\frac{1}{m}\right)}$.
- Valeur de $\frac{dy}{dt}$ au point d'inflexion : $\frac{dy}{dt} = amK\left(\frac{m+1}{e * m}\right)^{\left(1+\frac{1}{m}\right)}$.

Le comportement du modèle est illustré sur les Fig.1 et 2 ci-dessous. L'asymptote K est fixée à 100. Du paramètre a, inverse d'une constante de temps, dépend la vitesse à laquelle l'asymptote est atteinte (voir Fig.1). Le paramètre m « règle » la position du point d'inflexion (voir Fig.2).

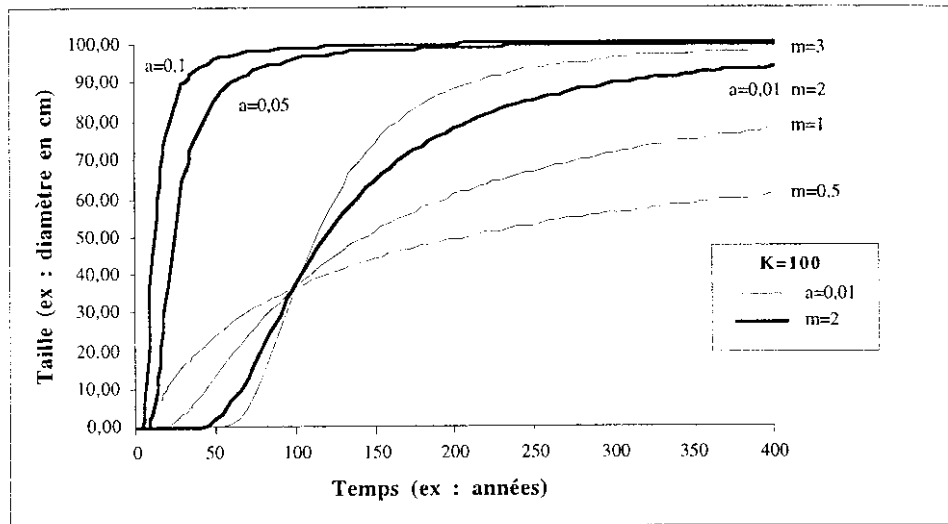


Fig.1. Illustration de la forme intégrale du modèle de Korf. La valeur de K est fixée à 100.

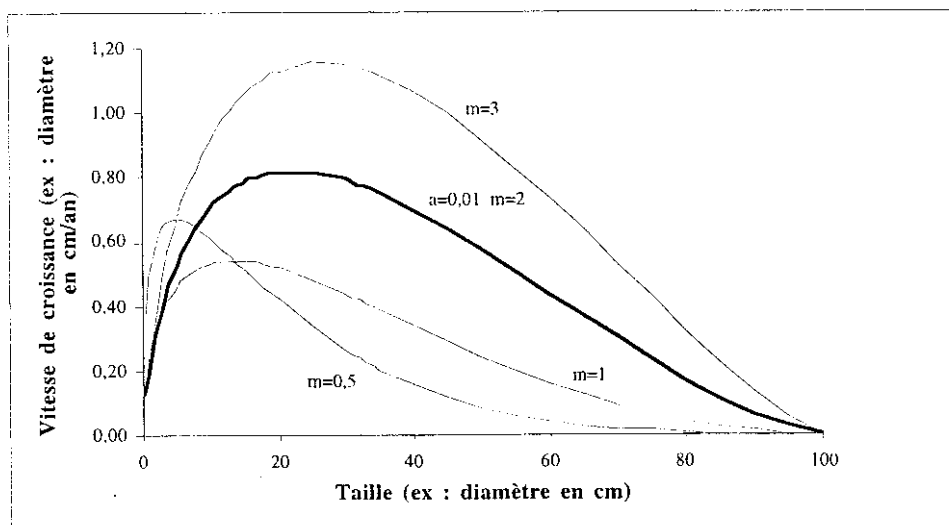


Fig.2. Illustration de la forme différentielle du modèle de Korf. La valeur de K est fixée à 100, la valeur de a est fixée à 0,01.

Modèle de Gompertz (cas limite du modèle de Korf quand $m \rightarrow \infty$)

- Equation analytique : $y(t) = Ke^{-e^{-a(t-t_0)}}$ ou $y(t) = K\left(\frac{y(0)}{K}\right)^{e^{-at}}$.
- $y(0) = Ke^{-e^{-at_0}}$.
- Equation différentielle : $\frac{dy}{dt} = ay[\text{Log}(K) - \text{Log}(y)]$.
- Position du point d'inflexion : $y = \frac{K}{e}$.
- Valeur de $\frac{dy}{dt}$ au point d'inflexion : $\frac{dy}{dt} = a\frac{K}{e}$.

Le comportement du modèle est illustré sur les Fig.3 et 4 ci-dessous. L'asymptote K est fixée à 100 et la valeur de y à l'origine : $y(0)=10$. Du paramètre a , inverse d'une constante de temps, dépend la vitesse à laquelle l'asymptote est atteinte (voir Fig.3).

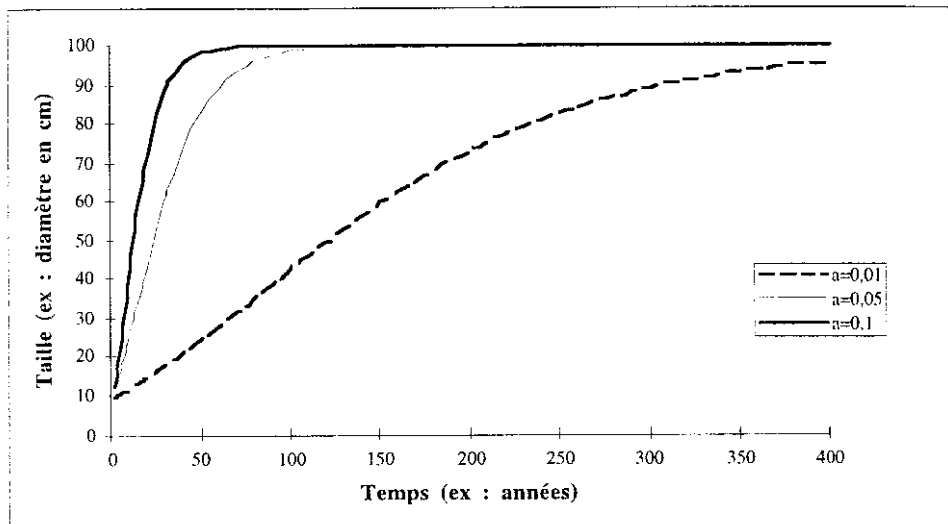


Fig.3. Illustration de la forme intégrale du modèle de Gompertz. $K = 100$ et $y(0)=10$.

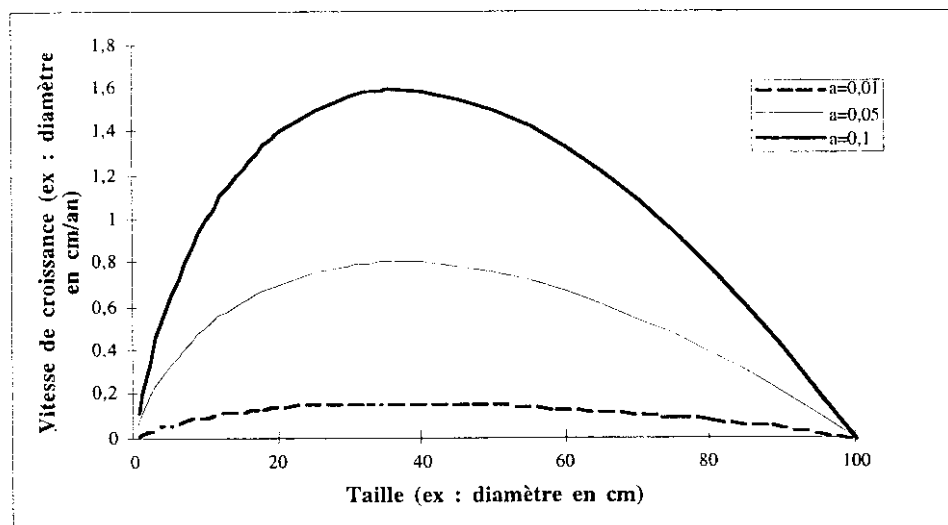


Fig.4. Illustration de la forme différentielle du modèle de Gompertz. $K = 100$ et $y(0)=10$.

ANNEXE 4

Article

Indices de compétition en forêt dense tropicale humide : étude de cas sur le dispositif sylvicole expérimental de Paracou (Guyane française)

Sylvie GOURLET-FLEURY

*CIRAD¹ -Forêt, Programme Forêt Naturelle - BP 701 - 97387 KOUROU Cedex, Guyane
française.*

(A paraître dans Annales des Sciences Forestières -1997)

Résumé – Dans le cadre de la mise au point, sur le dispositif sylvicole de Paracou (Guyane française), d'un modèle de croissance d'arbre dépendant des distances, nous avons passé en revue plusieurs indices de compétition utilisés en forêt tempérée depuis les années 50 et envisagé leur application au cas de la forêt dense tropicale humide. Quatre indices simples calculés sur des placettes circulaires centrées sur l'arbre sujet, ainsi que leur variation dans le temps, ont été retenus et leur efficacité vis-à-vis de la prédiction de l'accroissement en diamètre étudiée en fonction des variables suivantes : type d'intervention sylvicole, classe de diamètre et espèce à laquelle appartiennent les arbres sujets, rayon de la placette sur laquelle sont calculés les indices. Nous montrons que leur intérêt reste limité (toutes espèces confondues, nous n'avons pu "expliquer" plus de 21% de la variance observée sur l'accroissement moyen annuel en diamètre) ; les meilleurs résultats sont obtenus la plupart du temps avec un indice lié au statut social de l'arbre, pour les petites classes de diamètre, des placettes de grande taille (au moins 20 m de rayon) et certaines espèces à tempérament héliophile ou structurantes de la voûte forestière.

Titre abrégé : Indices de compétition en forêt tropicale

Mots-clés : indice de compétition / accroissement / voisinage / modèle / forêt dense tropicale humide

¹ Centre de coopération internationale en recherche agronomique pour le développement.

NOTATIONS UTILISEES DANS LE TEXTE

D : diamètre à 1m30, en cm.

ΔD : accroissement moyen annuel lissé sur 3 ans (cm/an).

ΔNBD : différence entre les deux valeurs de l'indice NBD, calculées respectivement en 1985 (avant traitement sylvicole) et en 1988 (après traitement sylvicole). $\Delta NBD = NBD_{1988} - NBD_{1985}$.

ΔNBT : différence entre les deux valeurs de l'indice NBT, calculées respectivement en 1985 (avant traitement sylvicole) et en 1988 (après traitement sylvicole). $\Delta NBT = NBT_{1988} - NBT_{1985}$.

ΔSTD : différence entre les deux valeurs de l'indice STD, calculées respectivement en 1985 (avant traitement sylvicole) et en 1988 (après traitement sylvicole). $\Delta STD = STD_{1988} - STD_{1985}$.

ΔSTT : différence entre les deux valeurs de l'indice STT, calculées respectivement en 1985 (avant traitement sylvicole) et en 1988 (après traitement sylvicole). $\Delta STT = STT_{1988} - STT_{1985}$.

IC : indice de compétition quelconque.

NBD : nombre de voisins présents sur une placette circulaire centrée sur l'arbre sujet, et dont le diamètre est supérieur ou égal au sien.

NBT : nombre total de voisins (arbres de plus de 10 cm de diamètre) présents sur une placette circulaire centrée sur l'arbre sujet.

R : rayon d'une placette circulaire centrée sur chaque arbre, utilisée pour identifier ses compétiteurs.

STD : surface terrière cumulée des voisins présents sur une placette circulaire centrée sur l'arbre sujet, et dont le diamètre est supérieur ou égal au sien.

STT : surface terrière cumulée de l'ensemble des voisins (arbres de plus de 10 cm de diamètre) présents sur une placette circulaire centrée sur l'arbre sujet.

INTRODUCTION

Contexte de l'étude

L'étude des phénomènes de compétition en forêt dense tropicale humide s'insère dans un cadre plus large de recherche sur les mécanismes qui régissent le fonctionnement de cet écosystème, afin de donner au gestionnaire forestier les moyens d'intervenir plus rationnellement dans les peuplements. Si, en forêt tempérée, il dispose pour cela d'une masse de connaissances et d'outils élaborés (Ek et Dudek, 1980, Houllier et al, 1991, Houllier, 1992), en forêt tropicale naturelle beaucoup reste à faire.

En Guyane française, la forêt dense ombrophile sempervirente de plaine couvre environ 90% des 8,5 millions d'hectares du département, et la gestion de la quasi-totalité de cette superficie est confiée à l'Office National des Forêts (ONF). Seule la partie nord du massif forestier, située le long de la côte sur une profondeur de 60 à 70 kilomètres (1,5 millions d'hectares), fait l'objet d'un aménagement forestier global.

La pratique des éclaircies, bien maîtrisée en zone tempérée, pose de nombreux problèmes en forêt dense tropicale humide, par exemple :

- i) quelles règles simples de sylviculture adopter dans des peuplements aussi hétérogènes ?
- ii) comment procéder aux éclaircies (abattage, annélation, empoisonnement des arbres que l'on souhaite éliminer au profit des autres) ?
- iii) quels seront les effets des interventions sur la dynamique des peuplements traités, sur leur composition floristique, sur les populations animales qu'ils abritent et dont ils dépendent, sur la qualité technologique des futurs arbres récoltés, etc. ?

De nombreuses études expérimentales ont été entreprises ou sont en cours pour tenter d'apporter une réponse à ces questions (voir synthèse dans les numéros 219 et 220 "Spécial Guyane" de Bois et Forêts des Tropiques, CTFT, 1990). Parallèlement à l'approche expérimentale, nécessitant d'importants moyens humains et financiers ainsi qu'un certain recul dans le temps, une approche basée sur la modélisation, qui permet de s'affranchir partiellement de l'ensemble de ces contraintes, est mise en oeuvre depuis 1992 au CIRAD-Forêt dans le domaine de la dynamique des peuplements forestiers, tant au niveau du peuplement (Favrichon, 1995) qu'au niveau de l'arbre (Gourlet-Fleury et Montpied, 1995). C'est ce dernier niveau qui nous intéresse ici plus particulièrement.

Objectifs

De nombreux éléments influencent la croissance d'un arbre (Assmann, 1970), notamment son patrimoine génétique et son accès aux ressources (lumière, eau ou nutriments) contrôlé en partie par la compétition. L'essentiel des efforts de quantification de cette compétition par des indices adaptés a porté sur les peuplements forestiers tempérés, et plus rarement sur les peuplements naturels tropicaux.

Nous passons d'abord en revue les grandes familles d'indices avant de faire un point rapide sur leur possibilité d'application en forêt dense tropicale humide. Puis nous présentons les conclusions d'une première étude exploratoire réalisée en 1991-1992 sur une parcelle du dispositif de Paracou, visant à identifier des indices pouvant participer à l'élaboration d'un modèle de croissance. Nous prolongeons ce travail en introduisant de nouvelles variables, dont l'efficacité pour la « prédiction » de la croissance des arbres sera testée de manière systématique sur nos données, toutes espèces confondues dans un premier temps, puis pour certaines espèces particulières choisies en fonction de leur tempérament écologique.

MATERIEL ET METHODES

Matériel

Conçu en 1982, le dispositif sylvicole de Paracou a été installé en 1984 sur la commune de Sinnamary à environ 50 km au nord-ouest de Kourou (Maître, 1982, Schmitt, 1984, 1985 et 1989). 12 parcelles de 9 hectares chacune, soit 108 hectares au total, ont été assises dans une zone de forêt intacte, après inventaire et étude pédologique préalables. 9 d'entre elles ont fait l'objet, entre 1986 et 1988, d'interventions sylvicoles d'intensité variable (3 types de traitement, appliqués chacun à 3 parcelles), les 3 dernières jouant le rôle de témoin. Les interventions ont été les suivantes :

- i) traitement 1 : exploitation forestière de type "traditionnel" pour le bois-d'oeuvre. Environ 10 tiges par hectare faisant plus de 50 à 60 cm de diamètre à 1m30 et appartenant à une cinquantaine d'espèces différentes ont été prélevées ;
- ii) traitement 2 : exploitation forestière pour le bois-d'oeuvre suivie d'une éclaircie par dévitalisation éliminant les arbres de plus de 40 cm de diamètre à 1m30, ne présentant pas d'intérêt commercial potentiel ;
- iii) traitement 3 : exploitation forestière pour le bois-d'oeuvre du même type que les précédentes, suivie d'une exploitation devant fournir du bois combustible (tiges d'espèces non commerciales faisant entre 40 et 50 cm de diamètre), et d'une éclaircie par dévitalisation (tiges d'espèces non commerciales faisant plus de 50 cm de diamètre).

Le tableau I présente, pour chaque traitement, l'évolution du nombre d'arbres présents et de la surface terrière correspondante entre 1984 et 1994.

Depuis 1984, le peuplement constitué de toutes les tiges de plus de 10 cm de diamètre à 1m30 situées sur les 6,25 hectares centraux de chaque parcelle (soit environ 46000 arbres au total) fait l'objet d'une mesure annuelle des circonférences par le CIRAD-Forêt. Tous les arbres sont numérotés, repérés par des coordonnées cartésiennes et la plupart sont identifiés botaniquement.

Méthodes

Quelques considérations sur les indices de compétition

De nombreux auteurs ont étudié les différents indices mis au point depuis les années 50 (Johnson, 1973 ; Ottorini, 1978 ; Daniels et al, 1986 ; Dreyfus, 1988 ; Gourlet-Fleury, 1992a, Biging et Dobbertin, 1992, parmi d'autres). Rappelons que les relations de "compétition" entre individus ne sont que l'un des nombreux aspects que peuvent revêtir les interactions entre arbres (voir Jacquard, 1968). La compétition se caractérise par "l'apparition d'effets déprimants entraînés par la mise en présence des organismes" (Malcom, 1966, cité par Jacquard, 1968). C'est essentiellement elle qui régule l'accès d'un individu aux ressources qui lui sont nécessaires, d'où l'intérêt d'essayer de quantifier ce phénomène le mieux possible. C'est la fonction que tentent de remplir les "indices de compétition", quelle que soit la famille à laquelle ils appartiennent : indices indépendants ou indices dépendants de la distance.

Indices indépendants de la distance

Leur calcul ne nécessite aucune connaissance sur la position des arbres du peuplement étudié et peuvent être construits avec des données issues d'inventaires classiques de terrain. Ils peuvent être basés sur la dimension des houppiers, traduisant le fait que les arbres ne se développent pas de la même manière en situation isolée et en peuplement : exemple du "Crown Competition Factor" de Krajicek et al (1961), des indices de Mitchell (1969, 1975) ou du "Crown Ratio" de Daniels et al (1975, cité par Daniels, 1976). Ou bien ils utilisent la surface terrière (ou le diamètre) et la hauteur des arbres voisins, de manière absolue ou relativement aux dimensions de l'arbre sujet : indices de Steneker et Jarvis (1963), "Basal Area Index" de Glover et Hool (1979), etc. A noter que dans la plupart des cas, les "compétiteurs", c'est-à-dire les voisins intervenant dans le calcul de l'indice, sont situés à l'intérieur d'une placette circulaire de rayon arbitrairement fixé, centrée sur le sujet. Mais l'ensemble du peuplement étudié peut également être pris en compte : Kohyama (1991, 1992a, 1992b) construit ainsi un « one-sided competition index », à partir de la surface terrière cumulée de tous les arbres du peuplement de taille supérieure à celle de chaque sujet.

Indices dépendants de la distance

Leur calcul fait cette fois intervenir la distance entre arbres, et nécessite de disposer de ces données, ou au moins d'une cartographie des individus. On peut distinguer 5 sous-familles d'indices de ce type (la nomenclature utilisée pour les 3 premières a été empruntée à Daniels et al (1986) :

- *Indices de chevauchement ("Overlap indices")*

Le choix des "compétiteurs" et l'expression de la compétition reposent ici sur le concept de "zone d'influence" d'un arbre. Celle-ci est en général circulaire, centrée sur l'arbre et censée représenter "la surface maximum d'étendue horizontale qu'il pourrait utiliser pour sa croissance" (Ottorini, 1978). Son extension est souvent reliée à l'extension maximale du houppier en situation de croissance libre, ou de manière plus ou moins empirique à différentes variables individuelles (diamètre, hauteur). Les compétiteurs sont alors les arbres dont les zones d'influence "chevauchent" celle du sujet, et l'importance de ces chevauchements est utilisée, de manière additive, pour quantifier la compétition subie par ce dernier. Une forme générale d'indice fréquemment utilisée est la suivante :

$$IC_i = \frac{b_1}{A_i} * \sum_{j=1}^{nc} a_{ij} * \left(\frac{E_j}{E_i} \right)^{b_2},$$

i et j désignant respectivement l'arbre sujet et ses compétiteurs, nc le nombre de compétiteurs de l'arbre i, A la superficie de la zone d'influence, a la superficie du chevauchement entre les zones d'influence et E un descripteur quelconque de l'arbre tel son diamètre (Bella, 1971). b₁ et b₂ sont des constantes.

De tels indices ont été développés par de nombreux auteurs (Staebler, 1951 et Newnham, 1964 cités par Johnson, 1973 ; Gerrard, 1967 ; Arney, 1973 cité par Smith et Bell, 1983 ; Bella, 1971 ; Ek et Monserud, 1974).

- *Indices utilisant les ratios taille/distance ("Size ratio distance indices")*

Le principe retenu dans la formulation de ces indices est le suivant : un arbre de gros diamètre éloigné du sujet exerce en théorie une pression de compétition équivalente à celle d'un arbre de petit diamètre, proche du sujet. On peut réunir ces indices sous une formule générale :

$$IC_i = b_1 * \sum_{j=1}^{nc} \frac{D_j^{b_2}}{D_i^{b_3}} * \frac{E_j}{L_{ij}^{b_4}}$$

i et j désignant respectivement l'arbre sujet et ses compétiteurs, nc le nombre de compétiteurs de l'arbre i, D le diamètre à 1m30, L la distance entre les arbres et E un descripteur quelconque de l'arbre tel le rayon de sa couronne (Hamilton, 1969), lorsqu'il ne vaut pas 1. b₁, b₂, b₃ et b₄ sont des constantes.

Les compétiteurs sont désignés de différentes manières : situés dans une placette circulaire de rayon fixé centrée sur le sujet (Steneker et Jarvis, 1963 ; Hamilton, 1969 ; Hegyi, 1974), ou bien désignés selon la méthode du relascope de Bitterlich (Spurr, 1962 ; Daniels, 1976).

- *Indices de polygones ("Polygon indices")*

Le concept utilisé ici est proche de celui de la "zone d'influence" : on fait une nouvelle fois appel à une représentation de l'espace utilisé par un arbre pour se développer ("Area Potentially Available" de Brown (1965), notamment cité par Moore et al (1973), "espace vital" de Bouchon (1969)). Cet "espace vital" a la forme d'un polygone entourant le sujet, et les compétiteurs sont les voisins qui participent à la construction du polygone : ses côtés sont perpendiculaires aux segments sujet/voisin, et situés à une distance variable, éventuellement bornée, du dit sujet (tessellation de Dirichlet ou pavage de Voronoï, Bouchon, 1979). La surface du polygone, ou son volume si l'on connaît la hauteur des arbres, sont considérés comme des indicateurs de la pression de compétition subie (voir Jack, 1967, Moore et al, 1973, Alemdag, 1978, Pelz, 1978, cité par Daniels et al, 1986, Daniels et al, 1986, Nance et al, 1987).

- *Autres indices*

Certains indices, plus complexes, sont construits en combinant plusieurs des méthodes évoquées précédemment : exemple du "zone count" de Opie (1968), indice de chevauchement pour lequel la zone d'influence de chaque arbre est déterminée par la méthode du relascope de Bitterlich, ou du "PAGAS" (Proportionally Available Growing-space Area of Sectors) de Alemdag (1978), faisant également appel au relascope et au système des polygones.

Hamilton (1969), Hatch et al (1975), Pretzsch (1992) entre autres ont construit des indices en "3 dimensions", calculés à l'aide d'équations de profil de houppiers assimilés à des formes géométriques simples (cônes, sphères, paraboloides) : ils expriment soit la "compétitivité" des sujets, soit la pression de compétition qu'ils subissent.

Il existe enfin des indices particuliers difficiles à rattacher aux familles précédentes, tels que l'"estimateur de Latham" (1972), qui utilise des couples stéréoscopiques de photographies aériennes, ou les indices calculés à l'aide de variogrammes (Bachacou et Décourt, 1976).

Efficacité des indices de compétition dans la prédiction de la croissance

Incorporés dans les modèles de dynamique forestière, les indices de compétition interviennent pour moduler la croissance potentielle des arbres et la mortalité. Leur efficacité, telle que rapportée par différents auteurs, est extrêmement variable, et dépend notamment de la nature de la variable croissance expliquée (accroissement en diamètre, en surface terrière, en hauteur, en volume, accroissement annuel ou sur plusieurs années) et du type de peuplement sur lequel ils sont testés (équienne monospécifique, inéquienne plurispécifique, jeune, âgé, dense, peu dense, etc.).

La plupart des auteurs travaillant sur des peuplements équiennes s'accorde à reconnaître que le meilleur prédicteur de l'accroissement, en diamètre ou en surface terrière, est le diamètre initial qui "explique" (au sens statistique) 30 à 80% de sa variabilité. Dans ce type de peuplement, le diamètre peut en effet être considéré comme une bonne mesure du statut social de l'arbre. L'adjonction d'un indice de compétition n'améliore pas énormément la précision du modèle, le plus souvent parce qu'il est, par construction, corrélé au diamètre initial du sujet (Gourlet-Fleury, 1992a).

Les indices généralement reconnus comme les plus efficaces appartiennent aux sous-familles des indices de chevauchement et de polygones, traduisant l'importance de la notion de "zone d'influence" ou d'"espace vital" autour d'un arbre et ce dans tous les types de peuplement (équiennes monospécifiques à inéquiennes plurispécifiques). Par ailleurs, la prise en compte de la taille respective des individus en présence est très importante (intervention du "statut social", qui peut être approché de manière qualitative — voir en particulier les travaux de Tomé et Burkhart, 1989 — ou quantitative, à partir des données de hauteur).

"Applicabilité" à la forêt dense tropicale humide

La grande majorité des études réalisées sur les indices de compétition concerne les forêts tempérées. Des milieux relativement simples, au regard de l'hétérogénéité rencontrée en forêt dense tropicale humide : diversité floristique (de 50 à 250 espèces arborescentes à l'hectare — voir Swaine et al, 1987, pour une comparaison de différents sites —) et hétérogénéité structurelle notamment. Quelques travaux testant et utilisant des indices de compétition en forêt tropicale peuvent cependant être mentionnés : Dawkins (1966, dans différentes forêts denses d'Afrique, d'Amérique latine et d'Asie) et Wadsworth et al (1988, en forêt dense de Puerto Rico) ont travaillé essentiellement sur la

dimension des sujets (diamètre initial, hauteur, largeur et intensité d'éclairement des houppiers) ou sur la densité globale ou locale des peuplements, dans leur recherche de prédicteurs de la croissance. Besse (1984) a testé quelques indices indépendants de la distance en Côte d'Ivoire. Wan Razali (1986) a utilisé des indices de densité globale (calculés sur des placettes de 0,4 ha) dans les forêts mixtes de Malaisie et a inspiré les travaux de Chai et Lemay (1993) sur les forêts marécageuses de ce pays. Toujours en Malaisie (partie occidentale de Sumatra), Kohyama (1991) a testé un indice global traduisant le statut social des individus. Favrichon (1991) a travaillé sur la liaison intensité d'éclairement des houppiers (codes de Dawkins) / croissance d'une espèce de forêt centrafricaine. Vanclay (1991) a eu recours à des indices de densité globale (placettes de 0,04 à 0,5 ha) dans l'étude des forêts du Queensland.

Dans ce type de milieu, les potentialités de croissance d'un individu vont a priori fortement dépendre de variables telles que :

- i) le type d'espèce à laquelle il appartient : "hard gambler" (Oldeman et Van Dijk, 1989), ou "espèce héliophile des strates intermédiaires et supérieures" (Lieberman et al, 1985, 1987), caractérisés par des diamètres moyens, une durée de vie courte et une croissance très rapide, à "hard struggler", ou "espèce de sous-bois ou des strates intermédiaires, tolérante à l'ombrage", caractérisés eux par des diamètres petits à moyens, de longues durées de vie et des croissances lentes, avec tous les intermédiaires entre ces deux extrêmes ;
- ii) son stade architectural (Hallé et Oldeman, 1970, Oldeman, 1974, Edelin, 1984) : appartenance à l'ensemble du passé, du présent, ou de l'avenir, traduisant en particulier son aptitude à réagir en croissance aux ouvertures du couvert ;
- iii) son statut social : en forêt tropicale, Dawkins (cité par Synnott, 1979) a mis au point une classification largement utilisée, mélangeant statut social et mode d'éclairement des houppiers. Il a obtenu des coefficients de corrélation linéaire allant de 0,40 à 0,47 entre ses codes et l'accroissement en diamètre des individus de certaines espèces de forêt dense africaine ; des résultats similaires ont été obtenus par Favrichon (1991) en République Centrafricaine.

Un certain nombre d'indices de compétition construits en pays tempéré sont difficilement transposables à ce type de forêt :

- i) soit pour des raisons techniques : certaines variables telles que hauteur totale des arbres ou dimensions des houppiers sont très difficiles à mesurer, du fait d'une visibilité le plus souvent réduite ;
- ii) soit pour des raisons fonctionnelles : assimiler les houppiers à des sphères ou des cônes, par exemple, risque d'être trop simplificateur dans des forêts où la quête de la lumière leur fait subir des déformations très importantes ; par ailleurs, les relations de commensalisme semblent suffisamment fréquentes pour rendre peu intéressants les indices trop sensibles à la distance entre arbres, de type "size ratio distance" (Gourlet-Fleury, 1992b).

On peut enfin noter que contrairement à ce que l'on observe dans les peuplements équiennes, le diamètre initial des arbres dans ce type de forêt ne constitue généralement pas un très bon indicateur de leur croissance ultérieure. Alder (1983) signale, pour différentes espèces de forêts tropicales humides des différents continents, que cette variable ne prend pas en compte plus de 20 à 40% de la variance observée sur les accroissements. Nous reviendrons plus bas sur ce résultat.

Application au dispositif sylvicole de Paracou

Seules 8 parcelles du dispositif (soit 2 par traitement) ont été utilisées dans notre étude, qui a porté sur les arbres présents et vivants de 1985 à 1991, ne présentant aucune anomalie dans les mesures et situés à une distance suffisante des bords des parcelles pour que des indices puissent être calculés sur des placettes circulaires de différent rayon. La prise en compte de l'ensemble des traitements nous a permis de disposer de types d'environnements très contrastés, plutôt fermés en parcelles témoins et parcelles exploitées et ouverts à très ouverts dans les parcelles éclaircies.

NB. Un récapitulatif des notations utilisées dans le texte qui suit figure en début d'article.

Choix de la variable "croissance" à prédire

Nous avons choisi de travailler sur l'accroissement en diamètre des arbres, lissé sur 3 années consécutives de manière à limiter l'influence des erreurs de mesure. Nous avons pour cela ajusté systématiquement, pour chaque arbre i , le modèle suivant :

$$D_{i,t} = a_i \cdot t + b_i + u_{i,t}$$

($D_{i,t}$ diamètre en cm de l'arbre i au temps t , a_i et b_i : paramètres à estimer, u_i écart entre valeur prédite et observée) aux 4 mesures prises en 1988, 1989, 1990 et 1991, soit après traitement sylvicole. $\Delta D_i = a_i$ représente l'accroissement moyen annuel en diamètre de l'arbre i , c'est la variable "dépendante" que nous voulons prédire à l'aide des indices de compétition.

Choix des indices de compétition

Les données réunies sur le dispositif rendent possible l'étude de nombreux indices, à l'exclusion de ceux qui sont basés sur des mesures de hauteur et de dimensions des houppiers. Une collecte de données supplémentaires réalisée sur une des parcelles en 1992 (Gourlet-Fleury, 1992b) a dans un premier temps permis de comparer :

- i) 5 indices indépendants de la distance : statut social, codes de Dawkins, nombre de voisins "gênants" (gêne appréciée subjectivement), nombre total de voisins de plus de 10 cm de diamètre (NBT) et surface terrière cumulée (STT) sur une placette circulaire de rayon R variant de 5 à 10 m, centrée sur l'arbre dont on étudiait l'accroissement en diamètre ;
- ii) et 4 indices dépendants de la distance (indices de Steneker et Jarvis, 1963), utilisant des ratios taille/distance.

Ce premier travail a montré :

- i) l'importance pour la croissance, toutes espèces confondues, de l'accès des houppiers à la lumière, la classification de Dawkins se révélant être la meilleure variable prédictrice (résultats conformes à ceux obtenus par Dawkins, 1966, et Favrichon, 1991) ;
- ii) l'inadaptation de certains indices utilisant des ratios taille/distance, ce qui s'explique sans doute en partie par l'existence de relations de commensalisme entre arbres poussant très proches les uns des autres (l'un des individus tire profit de la présence de l'autre, alors que sa proximité devrait représenter un gêne importante) ;
- iii) l'efficacité comparable d'indices simples du type NBT et STT et d'indices plus compliqués du type de ceux de Steneker et Jarvis ;
- iv) l'importance de l'effet "espèce". Les différences de comportement qui les caractérisent se traduisent dans la structuration verticale de la végétation : certaines espèces demeurent en sous-étage tandis que d'autres atteignent la voûte forestière. Cela influe fortement sur des variables telles que "statut social", "codes de Dawkins" et "nombre de voisins gênants". Par ailleurs, la sensibilité à la compétition varie vraisemblablement d'un type comportemental à l'autre et le même indice de compétition ne convient pas à toutes les espèces.

Nous avons voulu reprendre la recherche d'indices "efficaces" en utilisant cette fois des parcelles ayant subi tous les types de traitement sylvicole, ce qui permettait de trouver des arbres dans toutes les situations de compétition possibles. Nous avons retenu les indices les plus simples déjà étudiés exprimant une densité locale (NBT et STT) et décidé d'en rajouter d'autres prenant en compte de manière indirecte les problèmes de statut social dont l'importance a été soulignée plus haut, toujours sur des placettes de rayon R , à savoir :

- i) le nombre de tiges vivantes, de taille supérieure ou égale à celle du sujet (NBD) ;
- ii) la surface terrière correspondante (STD).

Ces derniers indices nous ont permis de nous affranchir du problème de l'attribution aux arbres de codes de Dawkins, information non disponible sur l'ensemble du dispositif.

Nous avons décidé par ailleurs de pallier le problème de l'aspect statique inhérent aux indices de compétition (critiqué notamment par Burton, 1993) en nous intéressant à leur évolution passée. On sait en effet que la mortalité, par

chablis ou sur pied, constitue le moteur principal de la dynamique des peuplements. De manière logique, lorsqu'un arbre disparaît, il libère de l'espace et des ressources pour ses voisins. A priori un arbre autour duquel la pression de l'environnement diminue (disparition d'arbres, entraînant une baisse de la valeur des indices de compétition), doit voir sa croissance stimulée. A l'inverse, il doit se trouver gêné en cas d'augmentation de la pression (recrutement, augmentation importante de la croissance des voisins, entraînant une augmentation de la valeur des indices). La prise en compte de cet historique a été faite au travers du calcul de l'évolution, entre 1985 et 1988, des 4 indices statiques retenus (valeur calculée en 1988), pour obtenir 4 indices "dynamiques" : ΔNBT , ΔSTT , ΔNBD et ΔSTD . La période 1985 / 1988 étant celle durant laquelle ont été effectuées les interventions sylvicoles, était également celle qui permettait d'obtenir la plus large gamme de valeurs des indices intéressés².

Un des problèmes pratiques posés par ces indices consistait en la détermination du rayon R des placettes permettant d'identifier les voisins intervenant dans le calcul. En théorie, ce rayon devrait correspondre à la distance maximale prospectée par l'arbre pour s'approvisionner en lumière, en eau et en éléments minéraux : on retrouve ici la notion d'"espace vital" évoquée plus haut. Nous avons donc systématiquement étudié des rayons allant de 10 à 40 m par pas de 5 m.

Le tableau II reprend la valeur moyenne de tous les indices étudiés sur les différentes parcelles retenues, pour un rayon de placette de 30 m. Il permet notamment de se faire une idée de l'impact des traitements sylvicoles. Par ailleurs, la distribution de quelques-uns d'entre eux est représentée fig.1.

Mesures de l'efficacité des indices

Nous avons recherché les indices les plus liés à l'accroissement étudié, en passant par le calcul de coefficients de corrélation linéaire simples et l'étude des résultats de régressions linéaires multiples. Parmi les problèmes qui se sont posés :

- i) l'hétéroscédasticité des résidus des régressions de type $\Delta D = f(IC, \Delta IC)$ (IC = indice de compétition), qui nous a conduit à travailler non pas sur la variable ΔD , mais sur sa transformée logarithmique $\text{Log}(\Delta D+c)$ (c : constante positive permettant de traiter les faibles croissances négatives, fréquentes dans ce type de forêt. Voir fig.2) ;
- ii) la non indépendance des individus étudiés en croissance : en toute rigueur, il n'aurait fallu travailler que sur des arbres présents sur des placettes non chevauchantes, mais cela aurait considérablement réduit le jeu de données : ainsi, pour des placettes de rayon 30 m, seuls 48 arbres auraient pu être retenus, ce qui interdisait notamment toute étude par classe de diamètre ou par espèce.

Nous avons calculé systématiquement, dans toutes les études de corrélation linéaire, le coefficient de rang de Spearman en plus du coefficient classique de Pearson. Les valeurs obtenues étant cependant très proches, et les conclusions en terme de significativité toujours concordantes, nous n'avons fait figurer sa valeur que dans le tableau III.

Chaque étude de corrélation linéaire entre accroissement et indices, que ce soit en fonction des traitements sylvicoles, des classes de diamètre ou des espèces, a été complétée par la recherche du meilleur modèle linéaire à deux variables (lorsque la prise en compte d'un deuxième indice avait un sens) $\text{Log}(\Delta D+c) = f(IC1, IC2)$, en faisant appel à des régressions pas à pas maximisant le coefficient de détermination³. Ce dernier, ainsi que l'identité de IC1 et IC2 figurent dans tous les tableaux concernés.

La recherche d'un rayon "optimal" de placette a été réalisée de deux manières :

- i) en comparant les coefficients de corrélation simple ou multiple accroissement / indices de compétition sur des placettes de taille croissante ;

² Donc, l'accroissement moyen annuel en diamètre est calculé entre 1988 et 1991, tandis que NBT, STT, NBD et STD sont calculés en 1988 soit en début de période de croissance et ΔNBT , ΔSTD , ΔNBD et ΔSTD sont calculés entre 1985 et 1988 (on respecte un pas de 3 ans).

³ Les indices de compétition sélectionnés à l'issue de cette étude, censés traduire l'influence de l'environnement, devaient intervenir par la suite dans la formulation d'un modèle de croissance plus complet. Nous avons donc délibérément limité la sélection à deux indices, de manière à ne pas alourdir le modèle final et conserver des expressions aisément manipulables et interprétables. Il faut noter par ailleurs que l'ajustement de ces modèles avait un but essentiellement exploratoire et que les hypothèses d'homoscédasticité et de normalité des résidus n'ont pas été systématiquement vérifiées : les résultats obtenus ont une valeur essentiellement indicative.

- ii) en recalculant les indices par tranches successives de 10 m, jusqu'à une distance de 50 m des sujets, selon le schéma de la fig.3a et en vérifiant que l'apport d'informations réalisé par chaque tranche successive possédait un intérêt réel par rapport à celui d'une placette initiale de taille 10 m. Nous avons pour cela comparé des modèles emboîtés selon le principe suivant : nous sommes partis du meilleur modèle linéaire à deux variables $\text{Log}(\Delta D+c) = f(\text{IC1}, \text{IC2})$ ajusté pour des placettes de rayon 10 m, puis nous avons inclus à ce modèle les variables IC1 et IC2 calculées pour la "tranche de distance" 10m-20m, en vérifiant au travers d'un test F l'utilité de cette inclusion. Et ainsi de suite jusqu'à 50 m.

Cette dernière opération a également été réalisée avec des tranches de superficie constante (donc de largeur décroissante : voir fig.3b), pour pallier le biais éventuel dû au fait que l'ajout à la placette initiale de tranches successives de largeur égale donc de superficie croissante, se traduisait par l'ajout d'un nombre, lui aussi croissant, d'arbres au stock de départ.

RESULTATS ET DISCUSSION

Influence du diamètre sur les accroissements

Toutes espèces confondues, sur les seules parcelles témoins, le coefficient de corrélation linéaire de Pearson entre le diamètre D des arbres et le logarithme de leur accroissement moyen annuel en diamètre sur 3 ans $\text{Log}(\Delta D+c)$ est faible : $r = 0,21$ ($p < 0,0001$). Ce résultat rejoint les observations de Alder (1983) et peut s'interpréter ainsi : les petits arbres ont certes une probabilité importante d'appartenir à une espèce de "sous-bois", d'être dominés, d'avoir un houppier très peu éclairé et donc une croissance lente, mais ce sont aussi ceux qui appartiennent souvent à "l'ensemble d'avenir" (*sensu* Hallé et Oldeman, 1970 : individus n'ayant pas encore subi de réitération et n'ayant pas atteint la taille maximale), et qui peuvent voir leur croissance augmenter de manière considérable dès que les conditions deviennent favorables. A l'inverse, les gros arbres qui font partie des ensembles structuraux supérieurs et ont toute "chance" de se trouver dans des conditions favorables (co-dominants ou dominants, houppiers bien éclairés) sont aussi ceux qui appartiennent aux ensembles architecturaux « du présent ou du passé » (toujours *sensu* Hallé et Oldeman, 1970 : individus ayant atteint leur taille maximale à la suite de nombreuses réitérations, sénescents, ou dégénérés à la suite d'un traumatisme) et dont la croissance ne va pas tarder à stagner.

Lorsque toutes les parcelles, témoins et traitées, sont prises en compte dans le calcul, le coefficient de corrélation linéaire chute à $r = 0,12$ ($p < 0,0001$). Dans un milieu qui change très peu et très lentement, comme c'est le cas en forêt naturelle intouchée, le diamètre d'un arbre peut être considéré comme un indicateur (certes peu efficace) de conditions de croissance passées ; ces conditions évoluant peu, il est normal que le diamètre soit également corrélé aux accroissements ultérieurs si la période considérée n'est pas trop longue. Par contre, dès qu'interviennent une exploitation et/ou un traitement sylvicole qui modifient soudainement et de façon importante l'environnement des arbres, le diamètre perd son potentiel de prédiction et la recherche d'autres indices prend toute son importance.

La variable D n'a par la suite plus été considérée, car elle est liée linéairement à l'indice NBD ($r = 0,84$, lorsque les placettes de calcul font 30 m de rayon).

Efficacité des différents indices étudiés, en fonction des traitements sylvicoles ou tous traitements confondus

Les résultats obtenus toutes espèces et tous diamètres confondus pour des placettes de rayon 30 m figurent dans le tableau III. Ils montrent que les indices indépendants du diamètre de l'arbre sujet (NBT, Δ NBT, STT, Δ STT) sont le plus souvent un peu moins performants que les autres, tout particulièrement dans les parcelles témoins où la densité est forte et la compétition pour la lumière importante. Les variations d'indice sont souvent des "prédicteurs" aussi efficaces que les indices eux-mêmes. A noter que tous ces indices sont plus ou moins liés entre eux, mais qu'en général, la liaison entre indice statique et indice dynamique correspondant est faible : $r \approx 0,2$ entre IC et Δ IC.

Etant données les périodes de calcul choisies pour les différentes variables, l'évolution des indices intègre les traitements sylvicoles et la croissance traduit la réaction du peuplement à ces traitements. Si aucune tendance nette ne

se dégage du côté des indices statiques on observe que, à quelques exceptions près, plus le traitement est fort moins les indices dynamiques paraissent efficaces et cette dégradation des résultats transparait clairement dans la valeur du coefficient de détermination. Cela signifie que la variation brutale de l'environnement des arbres dans les parcelles traitées est moins corrélée à leur réaction en croissance qu'une variation plus progressive en parcelles témoins. Cette observation pourrait s'expliquer par la réaction "négative" de certains sujets à une mise en lumière trop brusque et à une modification trop importante du microclimat local à la suite des traitements, tout particulièrement dans le n° 3.

Quel que soit le traitement sylvicole, et tous traitements confondus, le meilleur modèle linéaire permettant de prédire la croissance s'est révélé être :

$$\text{Log}(\Delta D+c) = a*\text{NBD} + b*\Delta\text{NBD} + d.$$

Ce résultat reste vrai pour toutes les tailles de placettes étudiées (voir tableau V). Pour des placettes de 30 m de rayon (10745 arbres dans le jeu de données), $a=-0,0031$, $b=-0,0104$, $d=-0,6992$ (coefficients significativement non nuls), et le modèle prend en compte 21% de la variabilité observée ($\hat{\sigma} = 0,356$). La constante c vaut 0,287, valeur choisie supérieure au plus petit accroissement négatif observé dans le jeu de données.

Influence du diamètre sur les liaisons indices de compétition / accroissement

Nous avons travaillé toutes espèces et tous traitements sylvicoles confondus, sur des placettes de rayon 10 m et 30 m, mais les tendances étant comparables nous n'avons fait figurer les résultats que pour les placettes de grande taille. Les arbres ont été ventilés en classes de diamètre de 5 cm d'amplitude, la dernière classe regroupant tous les individus de plus de 60 cm. Les coefficients de corrélation linéaire obtenus entre les différents indices de compétition et l'accroissement ont été reportés dans le tableau IV, et représentés sur les graphiques de la fig.4. Ils font apparaître que :

- i) la valeur des coefficients de corrélation décroît lorsque le diamètre des arbres sujets augmente, jusqu'à la classe 30-35 cm pour laquelle on observe les valeurs les plus faibles. Puis elle ré-augmente, avant de devenir non significative pour pratiquement tous les indices à partir de la classe 60 cm et plus. A noter qu'à partir de 50 cm de diamètre, les effectifs deviennent faibles dans les classes, et les résultats "erratiques" et peu fiables ;

- ii) les indices dépendant du diamètre (NBD et STD) ne sont plus significativement corrélés à l'accroissement à partir des classes de diamètre 50-55 cm (dès 40-45 cm pour les placettes de rayon 10 m), contrairement aux simples indices de densité locale (NBT, STT). STT donne le plus souvent de meilleurs résultats que les autres indices statiques. Par contre, dans pratiquement tous les cas, les variations d'indice ΔNBD et ΔSTD sont davantage liées à l'accroissement que ΔNBT et ΔSTT , ΔNBD étant de loin le plus "efficace".

La classe 30-40 cm constitue une classe "charnière" dans ce type de forêt. En effet, c'est dans cette gamme de taille que les arbres appartenant aux espèces des strates supérieures "passent" dans la voûte forestière, et que les arbres appartenant à de nombreuses espèces des strates inférieures atteignent leur taille maximale. La diversité des situations rencontrées est très grande : arbres dominés, co-dominants, dominants, de tempéraments divers, ce qui peut expliquer la mauvaise performance générale observée sur les indices.

Dans les classes plus petites, les arbres subissent une concurrence importante pour la lumière et sans doute également pour l'alimentation en eau et en substances nutritives : la performance des différents indices NBT, STT, NBD et STD est très comparable. La disparition, ou l'apparition d'individus "dominants" influe beaucoup sur la croissance ce qui explique l'efficacité de la variable ΔNBD . A noter que dans ces gammes de taille, les variables STT, STD d'une part et ΔSTT , ΔSTD d'autre part ont des valeurs très proches : leurs performances sont tout à fait comparables.

Dans les classes supérieures (à partir surtout de la classe 50-55 cm), la compétition pour la lumière devient en théorie beaucoup moins importante, alors qu'elle l'est toujours pour les autres facteurs de croissance. Les simples indices de densité locale donnent de meilleurs résultats que les indices NBD et STD, liés au statut social. Cependant, on observe toujours des arbres dominés ou co-dominants donc gênés dans leur accès à la lumière, dans ces gammes de diamètre (en moyenne, 0,5 "voisins" de taille supérieure pour des arbres de la classe 50-55 cm dans un rayon de 10 m, un peu

plus de 4 dans un rayon de 30 m) : la bonne performance des variables ΔNBD et ΔSTD peut s'expliquer par l'existence d'une forte capacité de réaction, grâce aux phénomènes de réitération (arbres "du présent", voir plus haut), à toute augmentation de la ressource lumineuse au niveau des houppiers. Cependant, au-delà de 55-60 cm de diamètre, les arbres ont, pour la plupart, perdu ces facultés d'adaptation et les variations d'indices ne sont plus liés à leur accroissement.

Le meilleur modèle linéaire à deux variables permettant de décrire la croissance fait pratiquement toujours intervenir ΔNBD , en association avec d'autres indices dynamiques : ΔNBT dans les petites classes de diamètre et ΔSTT dans les classes supérieures. C'est pour les petites classes de diamètre (de 10 à 30 cm) que ce type de modèle est le plus efficace.

Remarque sur les interactions. Les interactions évidentes existant entre le diamètre et les indices de compétition peuvent être prises en compte à travers l'introduction, dans le modèle $\text{Log}(\Delta D+c)=f(IC)$ d'un terme d'interaction $D*IC$. Si l'on recherche de manière systématique les deux variables les plus efficaces parmi 90 variables constituées de nos 8 indices de base et du diamètre, des monômes du deuxième degré correspondants et de tous les termes d'interaction pris deux à deux, ce sont l'indice NBD et le terme $D*\Delta NBD$ qui se retrouvent sélectionnés. Ce dernier terme traduit la nécessité, pour obtenir un effet positif sur les accroissements, d'éliminer d'autant plus d'arbres dans le voisinage d'un sujet que celui-ci est petit. Il traduit, à l'inverse, l'effet très négatif de l'apparition d'un individu dominant dans le voisinage d'un gros arbre (plus gênant que dans le voisinage d'un petit arbre). Le modèle utilisant ces deux variables est alors caractérisé par un coefficient de détermination de 0,24 ($\sigma = 0,348$).

Cependant, l'utilisation du terme $D*\Delta NBD$ dans un modèle plus complexe, traduisant à la fois les potentialités propres de l'individu, et l'effet régulateur de l'environnement perd de son intérêt au regard de la variable ΔNBD . Nous l'avons donc provisoirement écartée.

Recherche d'un rayon "optimum" pour les placettes de calcul

Evolution des liaisons indices de compétition/accroissement avec le rayon des placettes de calcul

Nous avons retenu 8609 individus situés à plus de 40 m des bords des 8 parcelles d'étude et fait varier le rayon des placettes entre 10 et 40 m par pas de 5 m. Les résultats toutes classes de diamètre et toutes espèces confondues figurent dans le tableau V, et sont représentés fig.5a et 5b. Sur cette même figure ont également été représentés pour comparaison les résultats obtenus sur les deux premières classes de diamètre.

Pratiquement tous les coefficients de corrélation augmentent avec la taille des placettes, jusqu'à 20 m de rayon, puis se stabilisent. On observe la même tendance, à une exception près (variable NBD pour la classe de diamètre 15-20 cm) en ne considérant que les petites classes de diamètre (voir fig.5c, 5d, 5e et 5f), pour lesquelles on note toujours la nette supériorité de la variable ΔNBD . Tous diamètres confondus, le meilleur modèle linéaire à deux variables fait intervenir NBD et ΔNBD dans tous les cas. Le coefficient de détermination se stabilise à 20% à partir de placettes de rayon 25 m.

Etude de l'apport de "tranches successives" à des placettes initiales de 10 m de rayon

L'un des problèmes posés par les résultats du tableau V est le suivant : les différences relativement faibles observées entre coefficients de corrélation sont-elles significatives, autrement dit est-il réellement indispensable de calculer les indices sur des placettes de rayon supérieur à 10 m ?

Pour répondre à cette question, seuls les deux indices les plus efficaces toutes classes de diamètre et toutes espèces confondues quelle que soit la taille de la placette, ont été pris en considération : soit NBD et ΔNBD . Nous avons étendu le rayon des placettes jusqu'à 50 m, ne conservant qu'un jeu de 6631 arbres tous situés à plus de 50 m des bords des 8 parcelles, et nous avons procédé à l'étude des modèles emboîtés : les résultats concernant des tranches successives de largeur constante figurent dans le tableau VI. Si l'on fixe le risque d'erreur à 5%, on constate que

chaque ajout d'information améliore le résultat global de manière très significative (particulièrement net en ce qui concerne le passage de 0-10m à 0-20m), et ce jusqu'à 50 m mais avec une valeur du F en diminution rapide. Si l'on inclut tous les indices dans le même modèle ($R^2 = 0,24$), on observe que jusqu'à 20 m les coefficients de NBD et de Δ NBD sont significativement non nuls, alors qu'au-delà, seule l'information apportée par l'indice dynamique Δ NBD reste utile. Les résultats obtenus en considérant des tranches successives de superficie constante sont tout à fait comparables aux précédents.

Comment expliquer ces différentes observations ? Il faut se souvenir que l'indice NBD est avant tout un indice de statut social. Un arbre dominé par son entourage proche l'est tout autant par son entourage lointain, de même un arbre dominant son entourage proche "domine" son entourage lointain : il n'est ainsi pas étonnant que les liaisons croissance/NBD varient peu avec l'éloignement, et qu'au-delà d'une certaine distance (en l'occurrence 20 m) l'information supplémentaire apportée par cette variable n'ait plus d'intérêt. En ce qui concerne Δ NBD, cet indice traduit une augmentation ou une diminution de la ressource (tous types de ressource, et particulièrement ressource lumineuse) dans l'entourage des arbres. Il semblerait que ceux-ci soient encore sensibles à des variations se produisant à grande distance, sans qu'il soit vraiment possible de trancher sur cette distance. Un seul fait paraît assez clair : ce rayon doit être supérieur ou égal à 20 m.

Stone et Kalisz (1991) ont recensé un grand nombre de travaux réalisés sur l'extension radiale des racines d'arbres isolés ou en peuplement, tant en forêt tempérée que tropicale : ils signalent que les valeurs les plus fréquemment rapportées pour cette variable se situent entre 10 et 20 m, mais que l'effet sur des cultures intercalaires ou avoisinantes atteint 30 m. En Guyane française, Atger et Edelin (1994) travaillant sur l'architecture des systèmes racinaires de différentes espèces notent une extension latérale dépassant 10 m chez *Cecropia obtusa* (espèce pionnière atteignant environ 30 cm de diamètre à 1m30) et Roggy (comm. pers.), a suivi les racines d'un *Sterculia pruriens* de 110 cm de diamètre sur plus de 50 m. A noter que nous avons toujours admis de manière implicite que l'extension racinaire des arbres, étroitement liée à la notion d'"espace vital", et à la superficie des placettes d'étude, était isotrope, ce qui est vraisemblablement faux et explique sans doute en partie que les résultats observés soient aussi peu concluants.

Influence de l'information "espèce" sur les liaisons accroissement / indices de compétition

Les études de ce type se heurtent rapidement au problème des faibles effectifs, dès que l'on considère séparément chaque espèce : le choix se trouve assez vite limité, lorsqu'en particulier on veut retenir des comportements contrastés. Il n'a pas été possible, par exemple, d'étudier d'espèces pionnières, trop peu abondantes dans les parcelles.

6 espèces ont été retenues : *Bocoa prouacensis* Aublet (Boco), *Carapa procera* A. de Candolle (Carapa), *Dicorynia guianensis* (Aubl) Amshoff (Angélique), première espèce exploitée en Guyane, *Licania micrantha* Miq, *Sloanea guianensis* (Aubl) Benth et *Symphonia globulifera* Linnaeus f, (Manil marécage). Leurs effectifs, accroissement moyen, diamètre moyen, diamètre maximum dans le fichier de données, ainsi que la valeur moyenne de l'indice NBD (très liée à celle du diamètre) pour chacune d'elles figurent au tableau VII (voir également fig.6a et 6b). Dans une classification basée essentiellement sur des critères de croissance et de structure diamétriques, Favrichon (1994) a rattaché ces espèces respectivement :

- i) *Bocoa prouacensis* et *Sloanea guianensis* au groupe 1 : "Espèces tolérantes (à l'ombrage) des étages inférieurs" ;
- ii) *Licania micrantha* au groupe 2 : "Espèces tolérantes de l'étage supérieur ;
- iii) *Dicorynia guianensis* et *Symphonia globulifera* au groupe 3 : "Espèces semi-tolérantes de l'étage dominant" ;
- iv) *Carapa procera* au groupe 4 : "Espèces héliophiles de l'étage supérieur".

Tous les diamètres ont été confondus, faute d'effectifs suffisants. Le calcul des coefficients de corrélation linéaire accroissement / indices et leur évolution avec le rayon des placettes font apparaître que :

- i) les indices de compétition les plus liés à l'accroissement dépendent de l'espèce, mais ce sont presque toujours des indices de statut social, et souvent des indices dynamiques (Δ NBD et Δ STD) sauf dans le cas de *Dicorynia guianensis*, très sensible à NBD ;

- ii) les indices statiques de statut social ne sont pas plus performants chez les espèces exigeantes en lumière que chez les autres. D'une manière générale, toutes les espèces étudiées (sauf *Dicorynia guianensis*) sont sensibles aux indices statiques bruts NBT et STT qui traduisent une compétition pour l'ensemble des ressources (et pas seulement la lumière) ;

- iii) il n'existe pas de gros écarts de performance entre les indices dynamiques, ce qui n'est pas le cas pour les indices statiques : ainsi, *Licania micrantha* et *Carapa procera*, deux espèces ayant pourtant des tempéraments très différents, sont beaucoup moins sensibles aux indices de statut social qu'aux autres l'inverse se produisant chez *Dicorynia guianensis*.

- iv) on notera enfin que chez l'espèce de petite taille *Sloanea guianensis*, tous les types d'indices, respectivement statiques et dynamiques, ont une efficacité comparable.

Il semble donc que les différentes espèces étudiées soient avant tout sensibles à une compétition exercée par l'ensemble de leurs voisins (prise en compte par les indices statiques NBT et STT), dans un rayon variable dépendant de leurs exigences physiologiques, et à la disparition ou l'apparition dans leur voisinage de sujets les dominant (prises en compte par les indices dynamiques Δ NBD et Δ STD).

Le comportement propre à chaque espèce transparait moins dans la nature des indices de compétition les plus liés à la croissance que dans leur efficacité. Le tableau VIII (voir aussi le graphique de la fig.7) l'illustre particulièrement bien : on y observe une bonne concordance entre degré d'exigence en lumière des espèces et valeur du coefficient de détermination du meilleur modèle linéaire à deux variables permettant de décrire leur accroissement. On notera que les deux indices quasi-systématiquement retenus dans ces modèles sont les mêmes d'une espèce à l'autre. Par contre, la taille des placettes permettant d'obtenir le meilleur résultat varie : de 15 à 30 m selon les cas. De manière assez logique, des placettes de petite taille paraissent mieux adaptées :

- i) chez les espèces de petite taille (*Sloanea guianensis*) ;

- ii) chez les espèces héliophiles (*Carapa guianensis*), sensibles à l'écran venant du voisinage proche ;

- iii) chez les espèces à croissance rapide (*Symphonia globulifera*) ayant besoin de mobiliser très vite des ressources.

Chez les autres espèces, relativement tolérantes, à croissance plus lente, des placettes de plus grande taille semblent mieux adaptées. Ces conclusions restent évidemment à valider au travers de l'étude d'un plus grand nombre d'espèces ; elles tendent cependant à montrer qu'il existe une marge d'amélioration de la qualité d'un modèle de croissance qui utiliserait des indices simples comme ceux que nous avons étudiés.

CONCLUSION

Dans le "meilleur des cas", les indices de compétition que nous avons utilisés à Paracou nous ont permis d'expliquer 47% de la variance observée sur l'accroissement moyen annuel en diamètre d'une espèce (*Carapa procera*). Toutes espèces confondues cette valeur tombe à 21%. Ces résultats sont conformes à ceux obtenus en forêt dense africaine par Dawkins (1966) et Favrichon (1991), et peuvent être comparés à ceux que donnent Chai et Lemay (1993) pour les forêts marécageuses mixtes du Sarawak (18% de la variance totale observée sur l'accroissement annuel en diamètre des arbres, toutes espèces confondues).

Si les indices de compétition peuvent donner une idée, certes grossière, de l'accès d'un arbre aux ressources en lumière, eau et éléments minéraux, la disponibilité de ces ressources constitue un facteur non négligeable d'explication de la croissance d'autant que l'on rencontre en forêt dense tropicale des situations très contrastées à grande échelle spatiale : variations rapides (sur quelques mètres) de la profondeur utile des sols, bas-fonds hydromorphes, poches de sable, etc. Mieux approcher cette question, mais aussi celles posées par la nature et l'importance des extensions racinaires, le tempérament des espèces, une meilleure identification des relations sociales

susceptibles de s'établir entre deux individus (faire en particulier la part entre relations de compétition et relations de commensalisme), paraissent être des voies à explorer pour améliorer notre compréhension de la croissance d'un arbre dans ce type de forêt.

Compte-tenu de ces limites, les deux indices NBD et Δ NBD (resp. effectif des voisins de taille supérieure à celle du sujet et variation dans le temps de cet effectif) calculés sur des placettes de rayon compris entre 20 et 30 m nous ont paru être les plus adaptés à une utilisation dans un modèle de croissance. Ils possèdent le gros avantage d'être robustes, simples à calculer et de ce fait aisément utilisables dans un simulateur.

REMERCIEMENTS

Je tiens à remercier François Houllier pour l'aide qu'il m'a apportée dans la réalisation de ce travail, ainsi que Jean Bouchon qui a bien voulu relire et critiquer ce texte.

L'étude a été réalisée dans le cadre d'un DEA et d'une thèse de doctorat financés par le CIRAD-Forêt, grâce au travail des nombreuses personnes qui, en Guyane française, alimentent la base de données de Paracou.

REFERENCES BIBLIOGRAPHIQUES

- Alder D (1983) Growth and Yield of Mixed Tropical Forests. Consultancy report prepared for the FAO
- Alemdag IS (1978) Evaluation of some competition indexes for the prediction of diameter increment in planted White Spruce. Forest Management Institute, Information Report FMR-X-108, 38 p
- Assmann E (1970). The principles of forest yield study. Pergamon Press, 506 p
- Atger C, Edelin C (1994) Premières données sur l'architecture comparée des systèmes racinaires et caulinaires. *Can J Bot* 72, 963-975
- Bachacou J, Décourt N (1976) Etude de la compétition dans des plantations régulières à l'aide de variogrammes. *Ann Sci For* 33, 177-198
- Bella IE (1971) A new competition model for individual trees. *For Sci* 17, 364-372
- Besse F (1984) Contribution à l'étude de l'influence de l'environnement sur la croissance du Ba en forêt de Côte-d'Ivoire. DEA de Biologie et Physiologie Végétales, Université de Nancy I / CTFT, 48 p
- Biging GS, Dobbertin M (1992). A comparison of distance-dependent competition measures for height and basal area growth of individual conifer trees. *For Sci* 38, 695-720
- Bouchon J (1969) Sur "l'espace vital" des arbres forestiers. Document de la Station de Sylviculture et de Production du CNRF - INRA, 69-09, 12 p
- Bouchon J (1979) Structure des peuplements forestiers. *Ann Sci For* 36(3) : 175-209
- Burton PJ (1993) Some limitations inherent to static indices of plant competition. *Can J For Res* 23: 2141-2152
- Chai FYC, Lemay VM (1993) Development and testing of diameter increment models for mixed swamp forests of Sarawak. *For Ecol Manage* 58, 51-64
- CTFT (1990) Spécial Guyane. *Bois et Forêts des Tropiques*, 219 et 220, 157 p et 105 p
- Daniels RF (1976) Simple competition indices and their correlation with annual loblolly pine tree growth. *For Sci* 22, 454-456

- Daniels RF, Burkhardt HE, Clason TR (1986) A comparison of competition measures for predicting growth of loblolly pine trees. *Can J For Res* 16, 1230-1237
- Dawkins HC (1966) The productivity of tropical high-forest trees and their reaction to controllable environment. Balliol College, Nuffield Foundation research fellowship 1960-1963, Commonwealth Forestry Institute, Oxford, 111 p + tab
- Dreyfus P (1988) Compétition et croissance dans de jeunes peuplements résineux. Aspects biologiques, écologiques et sylvicoles. Thèse d'écologie forestière, Université de Droit, d'Economie et des Sciences d'Aix - Marseille, Faculté des Sciences et Techniques de Saint-Jérôme
- Edelin C (1984) L'architecture monopodiale : l'exemple de quelques arbres d'Asie tropicale. Thèse de Doctorat d'Etat, Université des Sciences et Techniques du Languedoc, Académie de Montpellier, 243 p
- Ek AR, Dudek A (1980). Development of individual tree based stand growth simulators : progress and applications. *Workshop on Forest Succession*. - Mountain Lake - USA - June 8-14, 1980, 1-25
- Ek AR, Monserud RA (1974) Trials with program FOREST : growth and reproduction simulation for mixed species even- or uneven-aged forest stands. In : *Growth Models for Tree and Stand Simulation* (J Fries, ed) Royal College of Forestry, Stockholm, *Res. Notes* 30, 56-73
- Favrignon V (1991). Sur quelques relations entre la croissance des arbres et la structure du peuplement en forêt semi-décidue (République Centrafricaine). Mémoire du DEA « Ecosystèmes Forestiers Tropicaux », Université Paris VI, 40 p
- Favrignon V (1994) Classification des espèces arborées en groupes fonctionnels en vue de la réalisation d'un modèle de dynamique de peuplement en forêt guyanaise. *Rev Ecol (Terre et Vie)* 49, 379-403
- Favrignon V (1995) Modèle matriciel déterministe en temps discret - Application à l'étude de la dynamique d'un peuplement forestier tropical humide (Guyane française). Thèse de doctorat de l'Université Claude Bernard Lyon I / CIRAD, 252 p + ann
- Gerrard DJ (1967) Competition quotient : an index of the competitive stress affecting individual forest trees. Michigan State University, Department of Forestry, 65 p
- Glover GR, Hool JN (1979) A basal area ratio predictor of loblolly pine plantation mortality. *For Sci* 25, 275-282
- Gourlet-Fleury S (1992a) Indices de compétition : les possibilités d'application à la gestion en forêt dense tropicale humide. Mémoire bibliographique du DEA d'Analyse et de Modélisation des Systèmes Biologiques, I.A.S.B.S.E., Université Claude Bernard Lyon I / CIRAD, 30 p + ann
- Gourlet-Fleury S (1992b) Simulation d'éclaircies sur le dispositif sylvicole de Paracou (Guyane française) : recherche de méthodes d'intervention en forêt, après exploitation. Mémoire technique du DEA d'Analyse et de Modélisation des Systèmes Biologiques, I.A.S.B.S.E., Université Claude Bernard Lyon I / CIRAD, 34 p + ann
- Gourlet-Fleury S, Montpied P (1995) Dynamique des peuplements denses forestiers en zone tropicale humide : ébauche d'un modèle d'arbre à Paracou (Guyane française). *Rev Ecol (Terre et Vie)* 50, 283-302
- Hallé F, Oldeman RAA (1970) Essai sur l'architecture et la dynamique de croissance des arbres tropicaux. Collection de Monographies de Botanique et de Biologie Végétale, n°6. Masson, 178 p
- Hamilton GJ (1969) The dependence of volume increment of individual trees on dominance, crown dimensions, and competition. *Forestry* 42, 133-144
- Hatch CR, Gerrard DJ, Tappeiner JC (1975) Exposed crown surface area : a mathematical index of individual tree growth potential. *Can J For Res* 5, 224-228
- Hegyí F (1974) A simulation model for managing jack-pine stands. In : *Growth Models for Tree and Stand Simulation* (J Fries, ed) Royal College of Forestry, Stockholm, *Res. Notes* 30, 74-87

- Houllier F, Bouchon J, Birot Y (1991). Modélisation de la dynamique des peuplements forestiers : état et perspectives. *Rev For Fran* 53: 87-105.
- Houllier F (1992). Analyse et modélisation de la dynamique des peuplements forestiers. Applications à la gestion des ressources forestières. Mémoire présenté devant l'Université Claude Bernard-LYON I pour l'obtention de l'habilitation à diriger des recherches. 67 p
- Jack WH (1967) Single tree sampling in even-aged plantations for survey and experimentation. Congrès IUFRO, Munich, 379-403
- Jacquard P (1968) Manifestation et nature des relations sociales chez les végétaux supérieurs. *Oecologia Plantarum* 3, 137-168
- Johnson EW (1973) Relationship between point density measurements and subsequent growth of southern pines. *Bulletin of Auburn University*, Agricultural Experiment Station, 447, 45 p
- Kohyama T (1991) Simulating stationary size distribution of trees in rain forests. *Ann Bot* 68: 173-180
- Kohyama T (1992a) Density-size dynamics of trees simulated by a one-sided competition multi-species model of rain forest stands. *Ann Bot* 70: 451-460
- Kohyama T (1992b) Size-structured multi-species model of rain forest trees. *Functional Ecol* 6: 206-212
- Krajicek JE, Brinkman KA, Gingrich SF (1961) Crown competition - A measure of density. *For Sci* 7, 35-42
- Latham RP (1972) Competition estimator for forest trees. *Photogrammetric engineering* 38, 48-50
- Lieberman D, Lieberman M (1987) Forest tree growth and dynamics at La Selva, Costa Rica (1969-1982). *J Trop Ecol* 3, 347-358
- Lieberman D, Lieberman M, Hartshorn G, Peralta R (1985) Growth rates and age-size relationships of tropical wet forest trees in Costa Rica. *J Trop Ecol* 1, 97-109
- Maitre HF (1982) Projet de recherches sylvicoles sur les peuplements naturels en forêt dense guyanaise. Rapport interne, CTFT
- Mitchell KJ (1969) Simulation of the growth of even-aged stands of white-spruce. *Bulletin of Yale University School of Forestry* 75, 48 p
- Mitchell KJ (1975) Dynamics and simulated yield of Douglas-fir. *For Sci, Monograph* 17, 38 p
- Moore JA, Budelsky CA, Schlesinger RC (1973) A new index representing individual tree competitive status. *Can J For Res* 3, 495-500
- Nance WL, Grissom JE, Smith WR (1987) A new competition index based on weighted and constrained area potentially available. In : *Forest Growth Modeling and Prediction*, Minneapolis (USA), August 24-28, 1987, 134-142
- Oldeman RAA (1974) L'architecture de la forêt guyanaise. *Mémoires ORSTOM*, vol 73, 198 p + ann
- Oldeman RAA, Van Dijk J (1989) Diagnosis of the temperament of tropical rain forest trees. In : *Rain Forest Regeneration and Management* (Gomez-Pompa A, Whitmore TC, Hadley M, ed), UNESCO, *Man and the Biosphere Series*, vol 6, 21-63
- Opie JE (1968) Predictability of individual tree growth using various definitions of competing basal area. *For Sci* 14, 314-323
- Ottorini JM (1978) Aspects de la notion de densité et croissance des arbres en peuplement. *Ann Sci For* 35, 299-320
- Prothery N (1995) Recherches sylvicoles sur les peuplements naturels en forêt dense guyanaise - Présentation de quelques résultats d'ensemble 11 ans après la mise en place du dispositif. Rapport interne CIRAD-Forêt, 42 p + ann

- Pretzsch VH (1992) Wuchsmodelle für mischbestände als herausforderung für die waldwachstumsforschung. *Forstw Cbl* 111: 87-105
- Schmitt L (1984) Recherches sylvicoles sur les peuplements naturels en forêt dense guyanaise. Phase préliminaire : localisation du dispositif principal. Rapport interne, CTFT, 38 p
- Schmitt L (1985) Recherches sylvicoles sur les peuplements naturels en forêt dense guyanaise. Présentation des résultats issus de la première campagne de mesures (état du dispositif à l'origine). Rapport interne, CTFT, 43 p
- Schmitt L (1989) Etude des peuplements naturels en forêt dense guyanaise. Compte-rendu de mise en application des traitements sylvicoles sur le dispositif de Paracou. Rapport interne, CTFT, 51 p
- Smith SH, Bell JF (1983) Using competitive stress index to estimate diameter growth for thinned Douglas-fir stands. *For Sci* 29, 491-499
- Spurr SH (1962) A measure of point density. *For Sci* 8, 85-96
- Steneker GA, Jarvis JM (1963) A preliminary study to assess competition in a white spruce - trembling aspen stand. *Forestry Chronicle* 39, 334-336
- Stone EL, Kalisz PJ (1991) On the maximum extent of tree roots. *For Ecol Manage* 46, 59-102
- Swaine MD, Lieberman D, Putz FE (1987) The dynamics of tree populations in tropical forest : a review. *J Trop Ecol* 3, 359-366
- Synnott TJ (1979) A manual of permanent plot procedures for tropical rainforests. Department of Forestry, Commonwealth Forestry Institute, University of Oxford, *Tropical Forestry Papers n°14*, 67 p
- Tomé M, Burkhart HE (1989). Distance-dependent competition measures for predicting growth of individual trees. *For Sci* 35, 816-831
- Vanclay JK (1991) Aggregating tree species to develop diameter increment equations for tropical rainforests. *For Ecol Manage* 42, 143-168
- Wan Razali BWM (1986) Development of a generalized forest growth and yield modelling system for mixed tropical forests of peninsular Malaysia. PhD of the University of Washington, 177 p
- Wadsworth FH, Parresol BR, Colon JCF (1988) Tree increment indicators in subtropical wet forest. In : *Growth and Yield in Tropical Mixed/Moist Forests*, Kuala-Lumpur, 20-24 Juin 1988, 15 p

TABLEAUX

Tableau I. Effectifs totaux par hectare (arbres faisant plus de 10 cm de diamètre à 1m30) et surface terrière correspondante, dans chaque groupe de 3 parcelles correspondant aux différents traitements réalisés à Paracou. L'exploitation a eu lieu d'Octobre 1986 à Mai 1987, et la dévitalisation de Décembre 1987 à Janvier 1988. D'après Prothery (1995).

Année	Témoins		Traitement 1		Traitement 2		Traitement 3	
	N	G (m ²)	N	G(m ²)	N	G(m ²)	N	G(m ²)
1984	625	30.9	603	30.6	623	31.9	628	32.1
1985	626	30.9	606	30.7	626	32.0	626	32.0
1986	626	31.0	606	30.7	626	32.1	625	32.1
1987	622	30.8	512	25.0	523	25.5	442	21.3
1988	618	30.7	506	24.9	496	20.4	425	17.6
1989	615	30.7	508	25.0	488	19.3	423	17.1
1990	614	30.7	509	25.1	489	19.2	431	17.2
1991	616	30.8	521	25.4	504	19.4	449	17.3
1992	614	30.9	530	25.6	513	19.7	470	17.7
1993	611	30.9	533	25.8	525	20.1	487	18.1
1994	607	30.9	531	25.9	527	20.3	490	18.5

Tableau II. Valeur moyenne en 1988 des principaux indices de compétition étudiés sur 8 parcelles de Paracou, pour des placettes de rayon 30 m (10745 individus). Toutes espèces confondues.

Traitement	Témoïn		T1		T2		T3	
N° de la parcelle	1	6	7	9	5	10	4	12
Nombre d'arbres étudiés	1596	1561	1376	1344	1449	1177	992	1250
ΔD (cm/an) lissée	0,14	0,13	0,19	0,20	0,23	0,32	0,28	0,25
Valeur moyenne des indices sur la parcelle								
NBT	168,1	159,0	149,2	147,6	161,3	138,4	128,0	149,0
ΔNBT	-4,9	-1,9	-28,2	-28,8	-28,7	-34,2	-59,0	-40,7
STT (m2)	8,3	8,3	7,7	7,0	6,2	5,9	5,2	5,9
ΔSTT (m2)	-0,2	0,0	-1,4	-1,4	-2,4	-3,3	-3,8	-3,2
NBD	89,0	81,8	76,3	74,9	81,8	70,6	65,5	75,4
ΔNBD	-4,5	-2,8	-14,2	-14,7	-19,0	-21,7	-33,3	-23,8
STD (m2)	6,7	6,7	6,0	5,4	4,7	4,5	3,9	4,4
ΔSTD (m2)	-0,3	0,0	-1,2	-1,2	-2,3	-3,1	-3,4	-3,0

NB. ΔNBT , ΔSTT , ΔNBD et ΔSTD représentent la variation de l'indice correspondant (resp NBT, STT, NBD et STD) entre 1985 et 1988.

Tableau III. Effet des traitements sylvicoles sur la valeur des coefficients de corrélation entre quelques indices de compétition calculés sur des placettes de rayon 30 m, et le logarithme de l'accroissement moyen annuel en diamètre des arbres sujets (10745 individus) : on a fait figurer respectivement le coefficient de corrélation linéaire de Pearson, et le coefficient de corrélation de rang de Spearman entre parenthèses. Toutes les espèces sont confondues.

Traitement	NBT	STT	NBD	STD		
Témoïn	-0,12 (-0,12)	-0,14 (-0,11)	-0,28 (-0,29)	-0,26 (-0,25)		
T1	-0,23 (-0,22)	-0,20 (-0,18)	-0,22 (-0,22)	-0,21 (-0,23)		
T2	-0,20 (-0,21)	-0,12 (-0,13)	-0,21 (-0,21)	-0,18 (-0,19)		
T3	-0,23 (-0,23)	-0,22 (-0,22)	-0,25 (-0,24)	-0,26 (-0,25)		
Tous	-0,26 (-0,26)	-0,29 (-0,30)	-0,25 (-0,26)	-0,31 (-0,33)		
	Δ NBT	Δ STT	Δ NBD	Δ STD	R ^{2*}	$\hat{\sigma}$ **
Témoïn	<i>NS (NS)</i>	<i>NS (NS)</i>	-0,33 (-0,26)	-0,12 (-0,19)	0,24	0,30
T1	-0,11 (-0,11)	-0,17 (-0,17)	-0,18 (-0,14)	-0,21 (-0,20)	0,18	0,34
T2	-0,12 (-0,14)	-0,15 (-0,15)	-0,23 (-0,21)	-0,19 (-0,18)	0,17	0,41
T3	-0,10 (-0,09)	-0,15 (-0,15)	-0,09 (-0,08)	-0,15 (-0,15)	0,13	0,37
Tous	-0,25 (-0,26)	-0,28 (-0,30)	-0,29 (-0,32)	-0,30 (-0,32)	0,20	0,36

NB. NS : coefficient non significativement différent de 0 ($P \geq 0,05$). Les autres coefficients sont tous significativement non nuls ($P \leq 0,0001$).

* Coefficient de détermination, traduisant la part de variance du logarithme de l'accroissement expliquée par la prise en compte, dans un modèle linéaire, des deux variables "explicatives" indiquées en italique gras dans chaque traitement. ** Estimation de l'écart-type résiduel correspondant à ce modèle (Log(cm)).

Tableau IV. Influence de la classe de diamètre sur les coefficients de corrélation linéaire (Pearson) calculés entre les indices de compétition étudiés et le logarithme de l'accroissement moyen annuel en diamètre des arbres sujets. Rayon des placettes : 30 m (10745 individus). Toutes les espèces sont confondues.

Classes de diamètre (cm)	Effectif	NBT	Δ NBT	STT	Δ STT	NBD	Δ NBD	STD	Δ STD	R ^{2*}	$\hat{\sigma}$ **
10-15	4197	-0,32	-0,31	-0,34	-0,35	-0,32	-0,46	-0,34	-0,37	0,28	0,31
15-20	2407	-0,27	-0,24	-0,31	-0,31	-0,22	-0,53	-0,30	-0,36	0,36	0,32
20-25	1436	-0,14	-0,16	-0,21	-0,19	-0,20	-0,38	-0,21	-0,23	0,20	0,40
25-30	920	-0,21	-0,24	-0,27	-0,26	-0,22	-0,43	-0,25	-0,30	0,24	0,35
30-35	657	-0,16	-0,12	-0,21	-0,19	-0,22	-0,33	-0,20	-0,24	0,16	0,38
35-40	517	-0,20	-0,19	-0,27	-0,22	-0,23	-0,38	-0,23	-0,28	0,21	0,38
40-45	250	-0,21	-0,30	-0,29	-0,33	-0,27	-0,41	-0,25	-0,35	0,18	0,38
45-50	150	-0,24	-0,35	-0,35	-0,44	-0,33	-0,47	-0,31	-0,46	0,23	0,42
50-55	73	-0,34	-0,23	-0,33	-0,32	<i>NS</i>	-0,43	<i>NS</i>	-0,44	0,23	0,49
55-60	61	-0,33	-0,29	-0,37	-0,31	<i>NS</i>	-0,34	-0,27	-0,31	0,16	0,43
≥ 60	77	-0,26	<i>NS</i>	-0,26	<i>NS</i>	<i>NS</i>	<i>NS</i>	<i>NS</i>	<i>NS</i>	0,07	0,43
Tous diamètres	10745	-0,26	-0,25	-0,29	-0,28	-0,25	-0,29	-0,31	-0,30	0,20	0,36

NB. NS : coefficient non significativement différent de 0 ($P \geq 0,05$). Les autres coefficients sont tous significativement non nuls ($P \leq 0,0001$).

* Coefficient de détermination, traduisant la part de variance du logarithme de la croissance expliquée par la prise en compte, dans un modèle linéaire, des deux variables "explicatives" indiquées en italique gras dans chaque traitement.

** Estimation de l'écart-type résiduel correspondant à ce modèle (Log(cm)).

Tableau V. Influence du rayon des placettes sur les coefficients de corrélation linéaire (Pearson) calculés entre les indices de compétition étudiés et le logarithme de l'accroissement moyen annuel en diamètre des arbres sujets. Toutes classes de diamètre et toutes espèces confondues (8609 individus étudiés).

Rayon des placettes (m)	NBT	Δ NBT	STT	Δ STT	NBD	Δ NBD	STD	Δ STD	R ^{2*}	$\hat{\sigma}$ **
10	-0,21	-0,19	-0,22	-0,19	<i>-0,26</i>	<i>-0,25</i>	-0,25	-0,20	0,14	0,38
15	-0,24	-0,22	-0,27	-0,24	<i>-0,27</i>	<i>-0,28</i>	-0,30	-0,26	0,17	0,37
20	-0,24	-0,23	-0,28	-0,26	<i>-0,26</i>	<i>-0,28</i>	-0,31	-0,28	0,19	0,37
25	-0,25	-0,23	-0,28	-0,28	<i>-0,26</i>	<i>-0,29</i>	-0,31	-0,30	0,20	0,37
30	-0,25	-0,23	-0,28	-0,28	<i>-0,25</i>	<i>-0,28</i>	-0,31	-0,30	0,20	0,37
35	-0,25	-0,24	-0,28	-0,28	<i>-0,25</i>	<i>-0,28</i>	-0,31	-0,30	0,20	0,36
40	-0,25	-0,24	-0,29	-0,28	<i>-0,25</i>	<i>-0,28</i>	-0,31	-0,30	0,20	0,36

NB. Les coefficients sont tous significativement non nuls ($P \leq 0,0001$).

* Coefficient de détermination, traduisant la part de variance du logarithme de la croissance expliquée par la prise en compte, dans un modèle linéaire, des deux variables "explicatives" indiquées en italique gras dans chaque traitement.

**Estimation de l'écart-type résiduel correspondant à ce modèle (Log(cm)).

Tableau VI. Comparaison de modèles emboîtés. A chaque étape, on ajoute au modèle linéaire global les deux indices NBD et Δ NBD correspondant à la "tranche" de distance suivante, et on teste l'utilité de cet apport d'information. Toutes classes de diamètre et toutes espèces confondues (6631 individus étudiés).

"Tranche" de distance (m)	Indices pris en compte	R ² du modèle	F*	Coefficient des indices dans le modèle complet**
0-10	NBD	0,15		S
	Δ NBD			S
10-20	NBD	0,20	224,6	S
	Δ NBD			S
20-30	NBD	0,22	68,5	NS
	Δ NBD			S
30-40	NBD	0,23	45,5	NS
	Δ NBD			S
40-50	NBD	0,24	29,2	NS
	Δ NBD			S

* $F_{lim}(2, \infty)_{ddl} = 3,00$ pour un risque d'erreur de 5%.

** S : valeur significativement différente de 0 ($P \leq 0,0001$), toujours négative. NS : valeur non significativement différente de 0 ($P \geq 0,05$).

Tableau VII. Caractéristiques moyennes, dans le fichier de données (8 parcelles, individus situés à plus de 40 m des bords), des 6 espèces retenues. NBD est calculé pour des placettes de rayon 30 m.

Espèce considérée	Effectif	Accroissement moyen annuel en diamètre (cm/an)	Diamètre moyen (cm)	Diamètre maximum (cm)	Valeur moyenne de NBD
<i>Bocoa prouacensis</i>	191	0,10	22,2	49,5	68,9
<i>Carapa procera</i>	130	0,37	22,7	61,3	61,9
<i>Dicorynia guianensis</i>	104	0,40	29,7	68,6	46,8
<i>Licania micrantha</i>	316	0,20	23,2	51,2	58,3
<i>Sloanea guianensis</i>	316	0,15	14,9	35,8	105,8
<i>Symphonia globulifera</i>	180	0,47	22,2	69,1	64,5

Tableau VIII. Meilleurs modèles linéaires à deux variables permettant de "décrire" l'accroissement moyen annuel en diamètre (sous la forme $\text{Log}(\Delta D+c)$) des 6 espèces étudiées.

Espèce considérée	Groupe d'espèce*	Rayon de la placette	Indices intervenant dans le modèle**	R ²	$\bar{\sigma}$
<i>Sloanea guianensis</i>	1	15 m	Δ NBD, NBD	0,21	0,23
<i>Bocoa prouacensis</i>	1	30 m	Δ NBD, (NBD)	0,23	0,23
<i>Licania micrantha</i>	2	25 m	NBT, Δ STD	0,27	0,25
<i>Symphonia globuli.</i>	3	20 m	Δ NBD, NBD	0,30	0,33
<i>Dicorynia guianensis</i>	3	30 m	NBD, (Δ NBD)	0,38	0,31
<i>Carapa procera</i>	4	15 m	Δ NBD, NBD	0,47	0,30

* Selon la classification de Favrichon (1994). Groupe 1 : "Espèces tolérantes des étages inférieurs". Groupe 2 : "Espèces tolérantes de l'étage supérieur". Groupe 3 : "Espèces semi-tolérantes de l'étage dominant". Groupe 4 : "Espèces héliophiles de l'étage supérieur".

** Les indices entre parenthèse ne sont pas liés linéairement à $\text{Log}(\Delta D+c)$ de manière significative. NB. Les coefficients des indices dans le modèle sont toujours négatifs.

LEGENDE DES FIGURES

Fig 1. Distribution de quelques indices de compétition calculés sur des placettes de rayon 30 m : (a) nombre total de voisins en 1988 ; (b) évolution de ce nombre entre 1985 et 1988 ; (c) nombre de voisins de diamètre supérieur à celui du sujet en 1988 ; (d) évolution de ce nombre entre 1985 et 1988. (e) Distribution des arbres sujets en classes de diamètre, en 1988. **NB.** 10745 individus sont pris en compte.

Fig 2. Accroissement moyen annuel (lissé sur 3 ans, entre 1988 et 1991) sur les 8 parcelles d'étude, représenté en fonction de deux indices de compétition calculés sur des placettes de 30 m de rayon : nombre de voisins dominants (a), et évolution du nombre des voisins dominants entre 1988 et 1991 (b). Même chose pour respectivement (c) et (d) en prenant $\text{Log}(\text{accroissement} + 0,287)$ en ordonnée. « Mini-4 » (resp. « Maxi-4 ») désigne la valeur de l'accroissement la plus faible (resp. la plus élevée) une fois que les 4 points extrêmes ont été éliminés.

Fig 3. Tranches successives utilisées pour la comparaison de modèles emboîtés. (a) Amplitude constante. (b) Superficie constante.

Fig 4. Influence de la classe de diamètre sur les coefficients de corrélation linéaire entre différents indices de compétition calculés sur des placettes de rayon 30 m et le logarithme de l'accroissement moyen annuel en diamètre des arbres sujets. Toutes les espèces sont confondues.

Fig 5. Influence du rayon des placettes de calcul sur les coefficients de corrélation linéaire (Pearson) entre différents indices de compétition et le logarithme de l'accroissement moyen annuel en diamètre des arbres sujets, toutes espèces confondues. (a) et (b) Tous diamètres confondus. (c) et (d) Arbres de la classe de diamètre 10-15 cm. (e) et (f) Arbres de la classe de diamètre 15-20 cm.

Fig 6. Caractéristiques moyennes des différentes espèces étudiées (pour les individus présents dans le fichier). (a) Représentation de la liaison indice NBD / diamètre (NBD est calculé sur des placettes de rayon 30 m). (b) Représentation de la liaison diamètre maximum / accroissement moyen annuel (lissé sur 3 ans).

Fig 7. Comparaison, pour différentes espèces, des coefficients de détermination obtenus avec le modèle linéaire $\text{Log}(\text{accroissement} + 0,287) = aIC_1 + bIC_2 + c$. IC_1 et IC_2 sont les deux indices de compétition qui maximisent les R^2 pour chaque espèce.

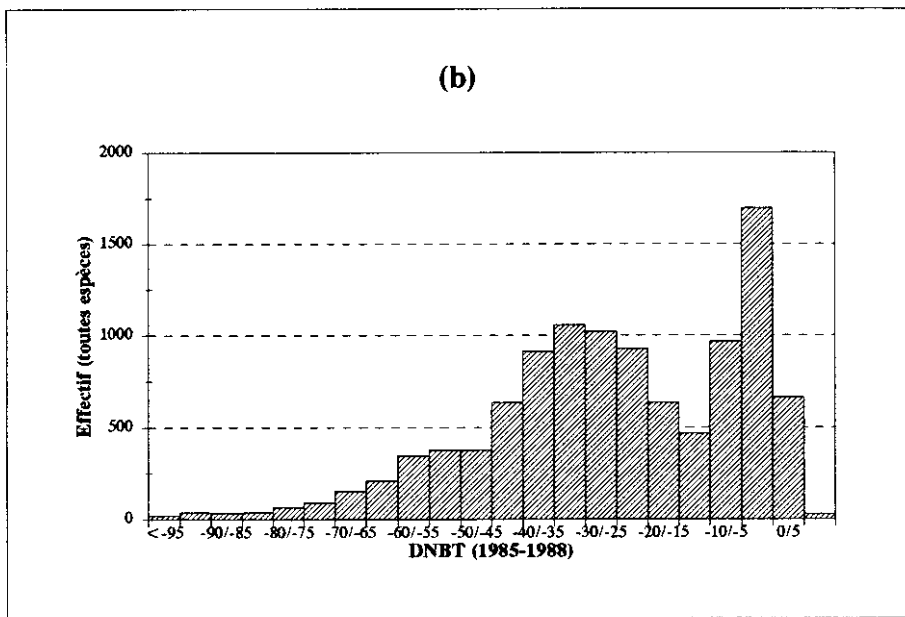
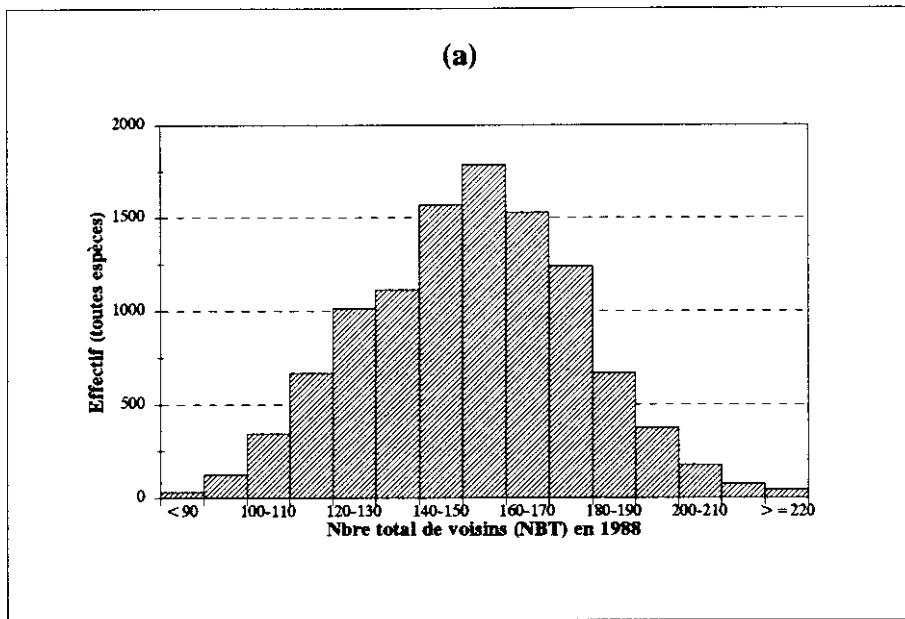


Fig 1.

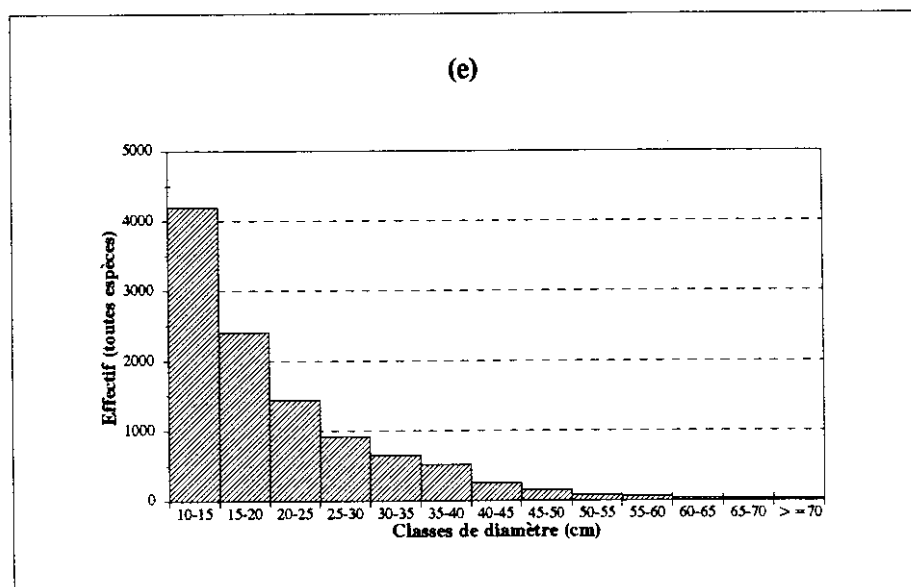
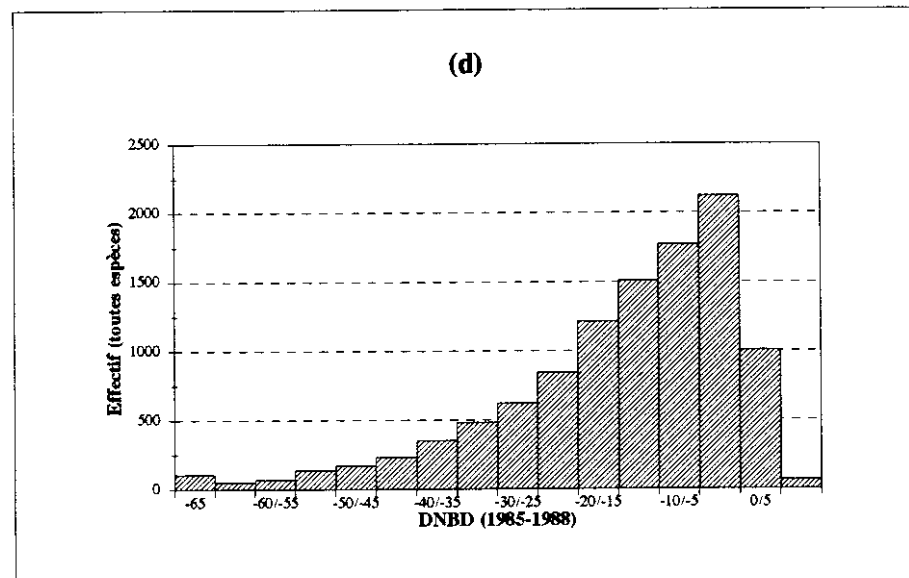
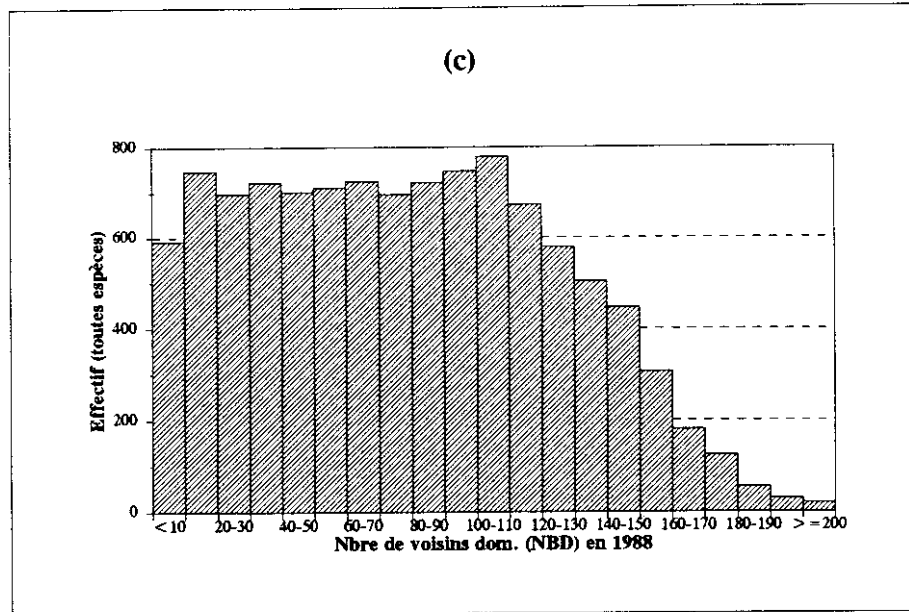


Fig 1 (bis).

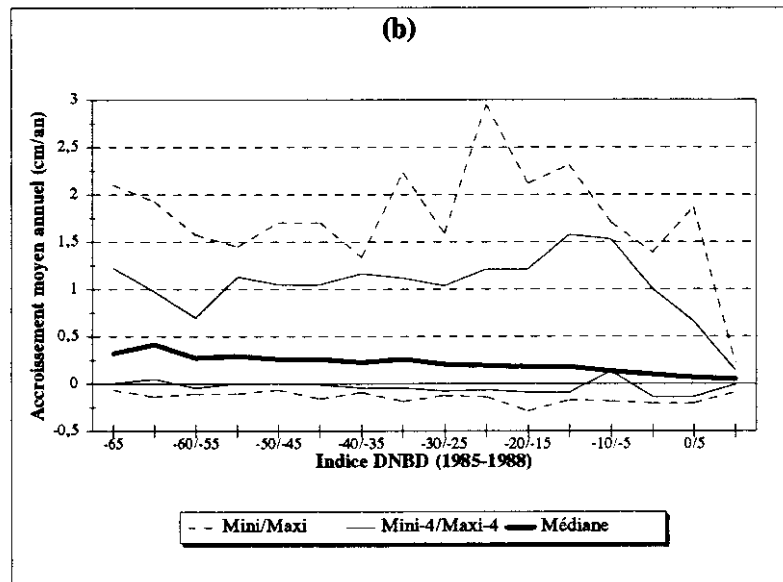
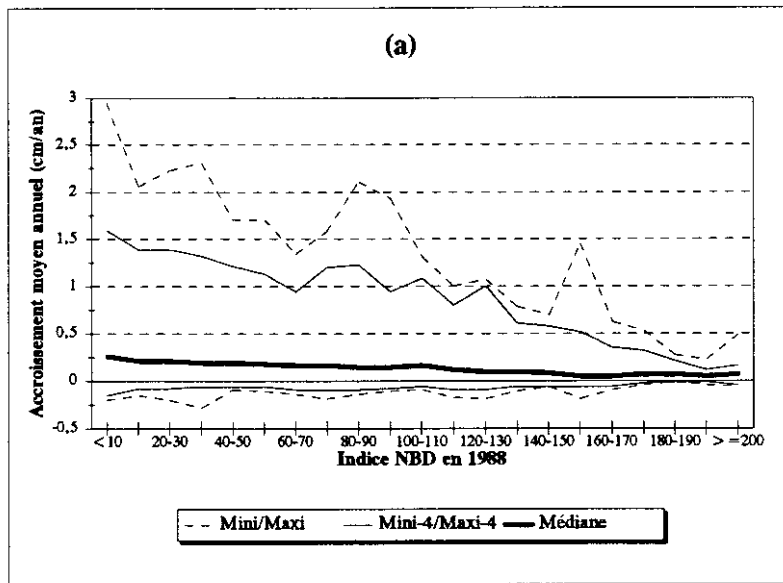


Fig 2.

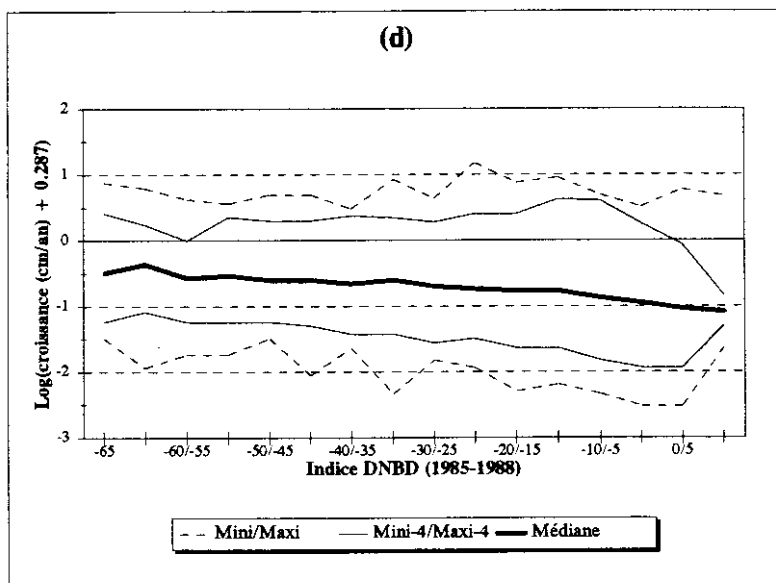
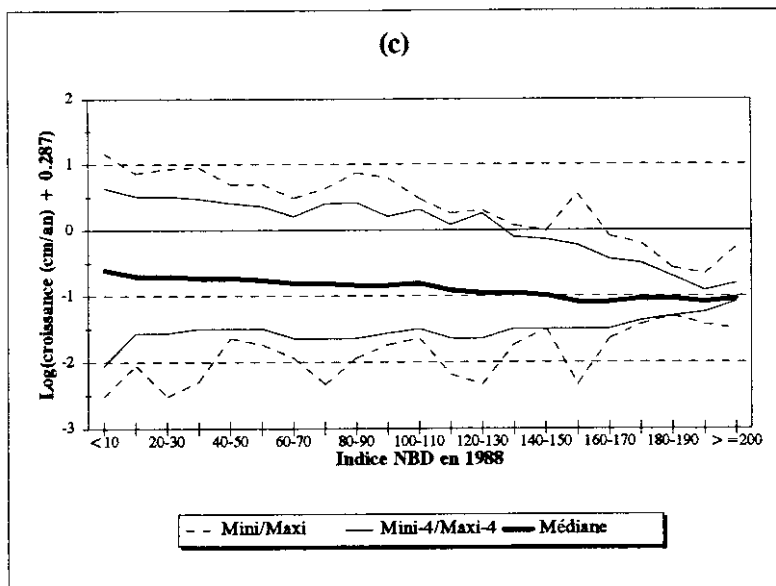


Fig 2 (bis).

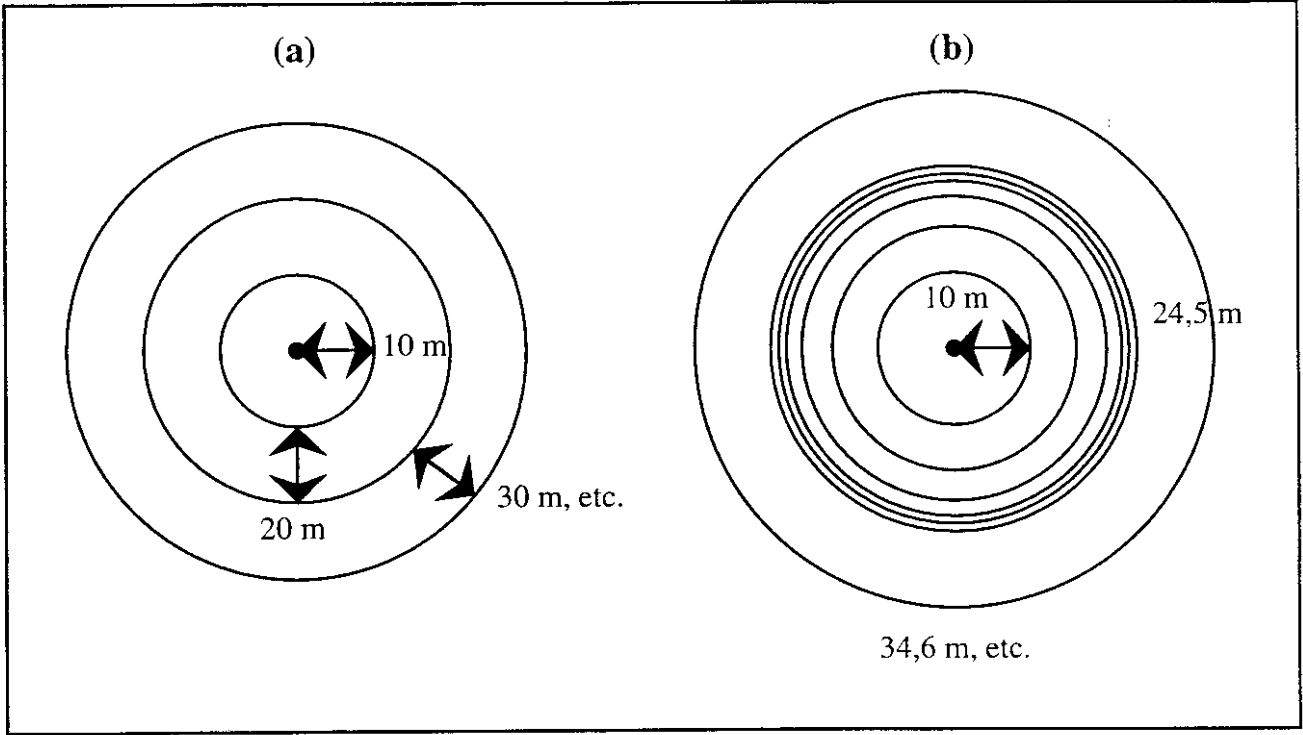


Fig 3.

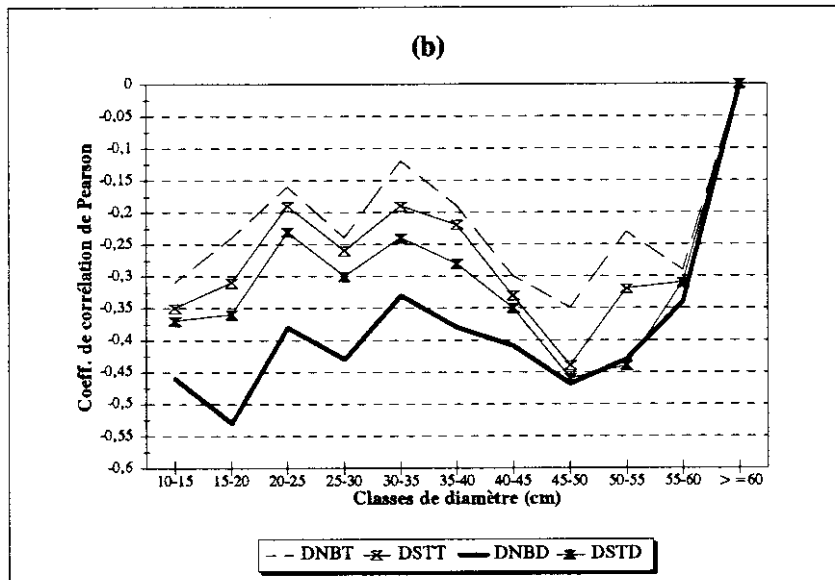
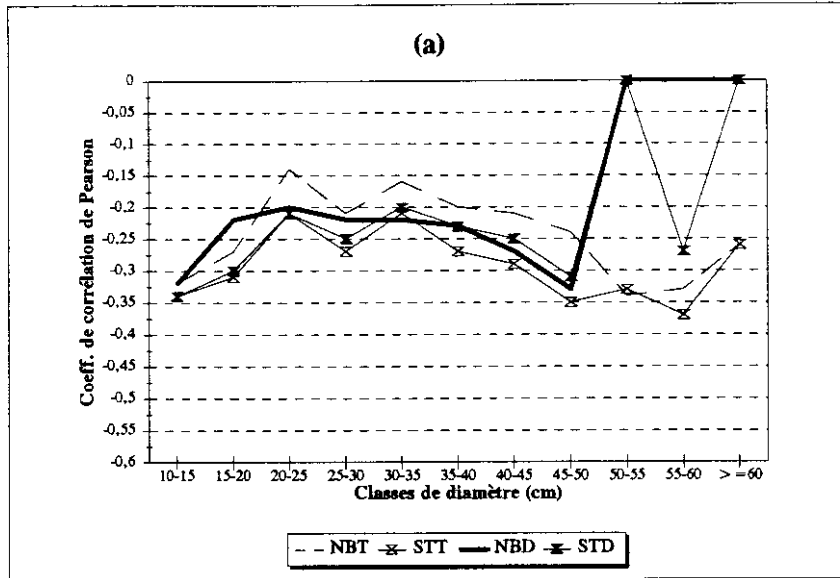


Fig 4.

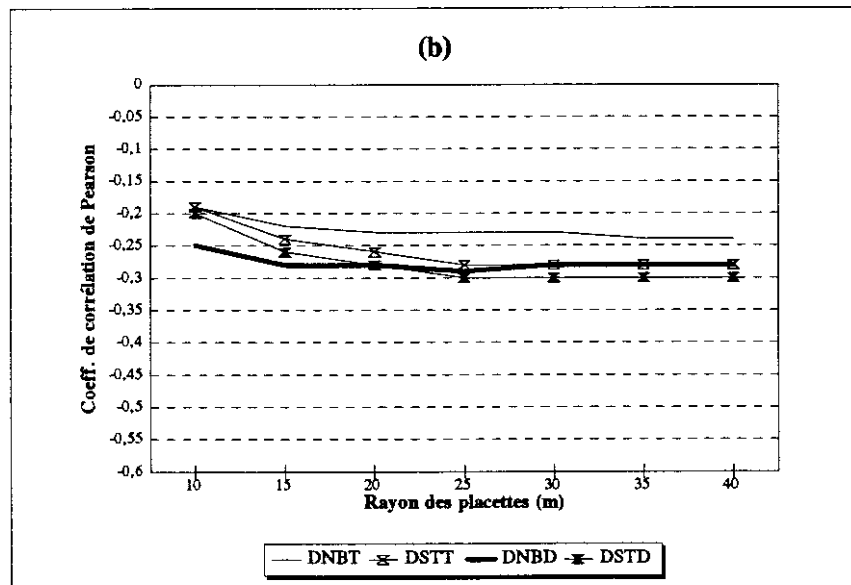
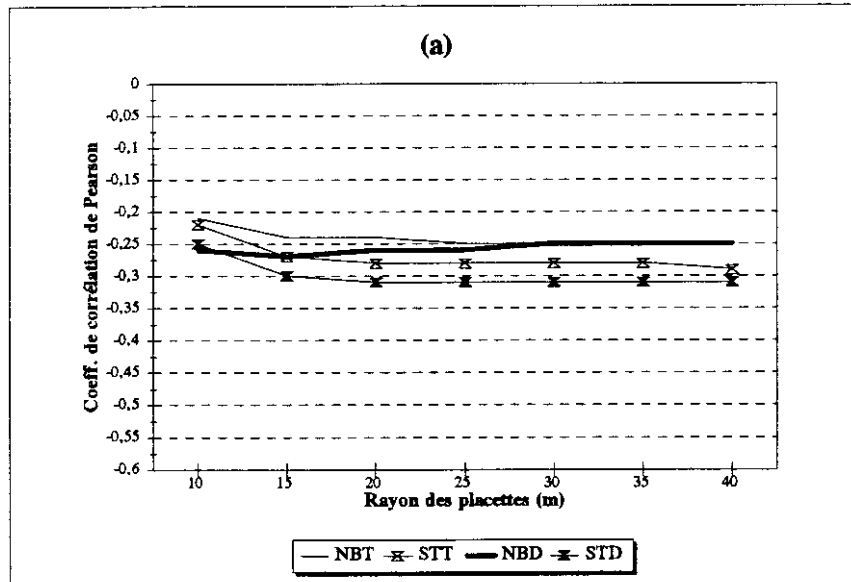


Fig 5.

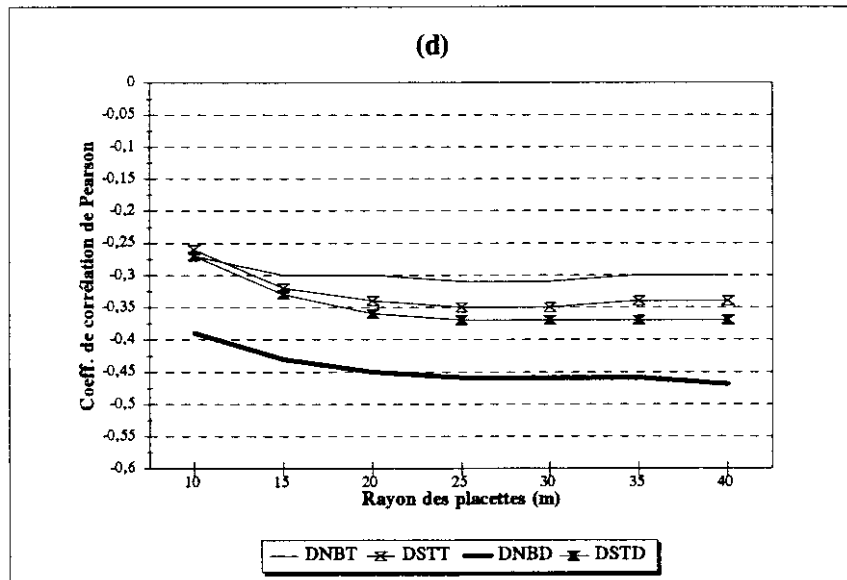
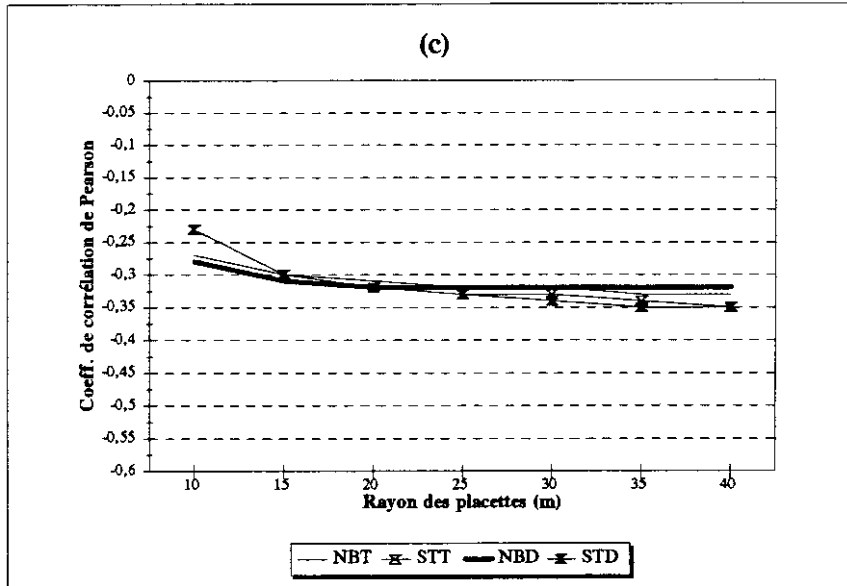


Fig 5 (bis).

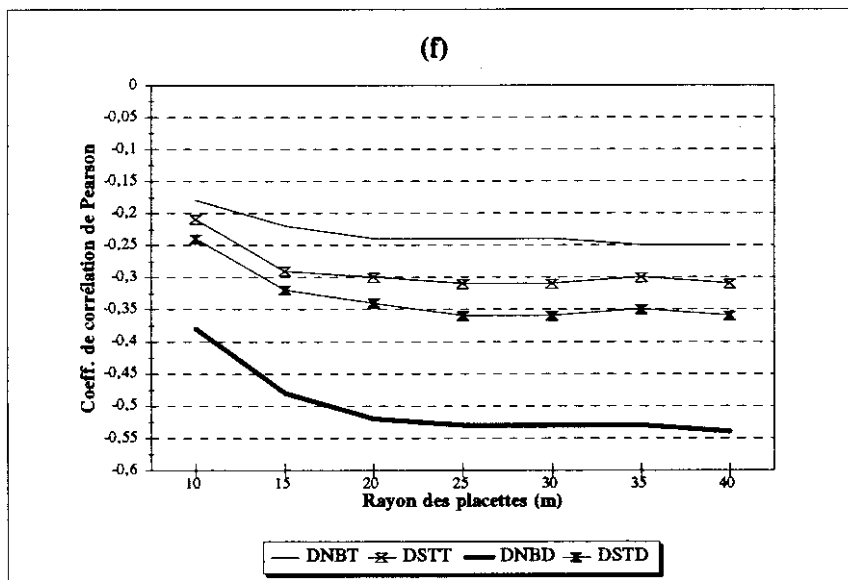
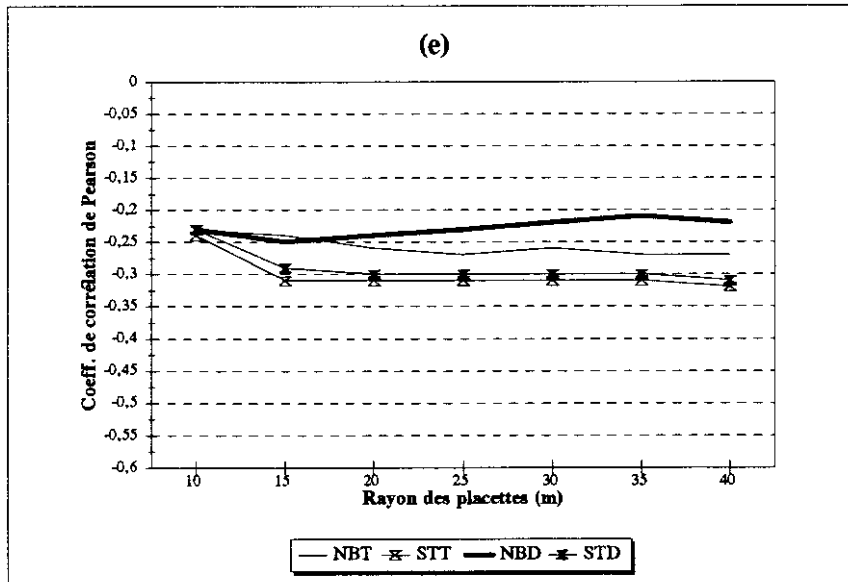


Fig 5 (ter).

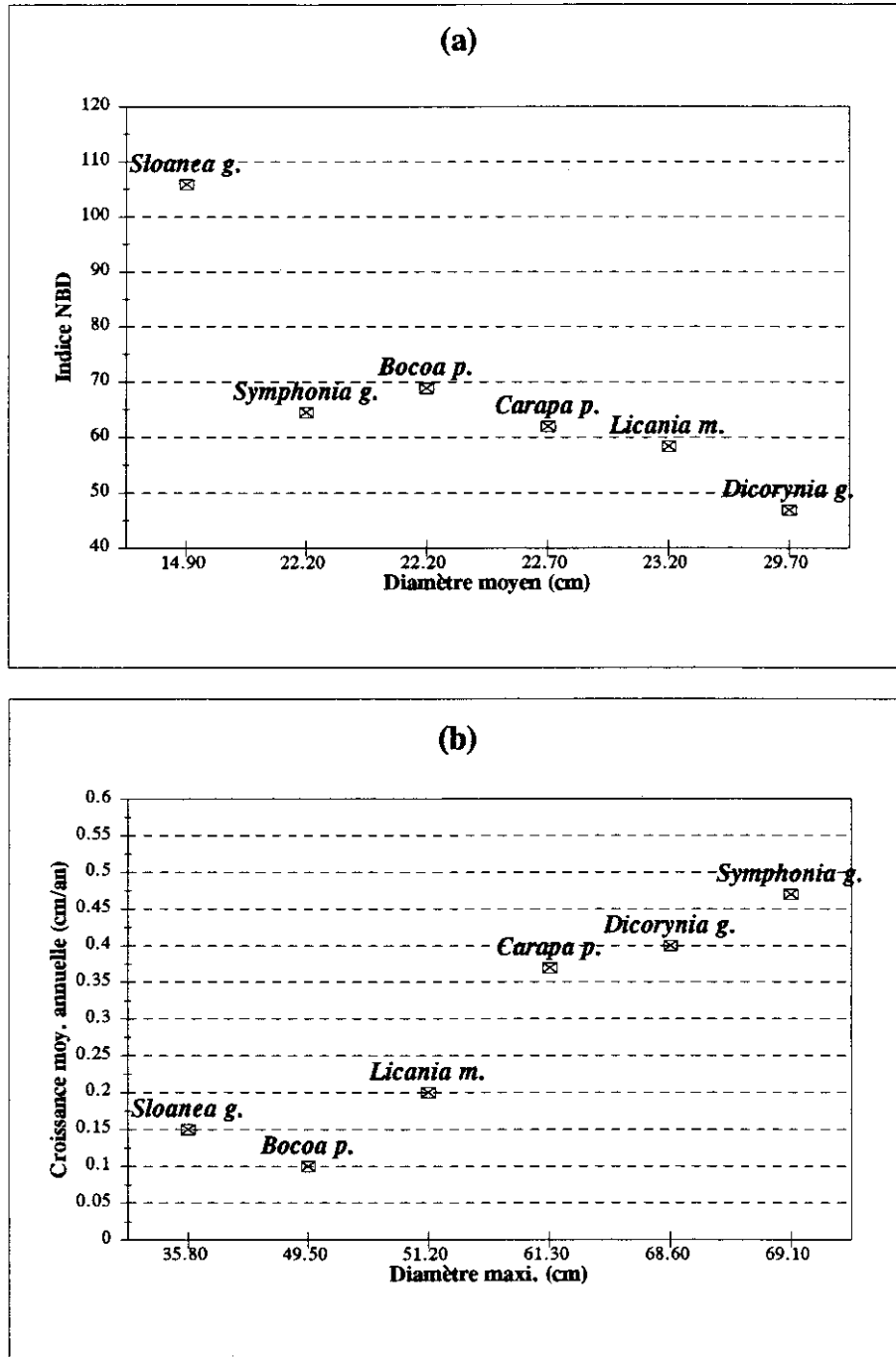


Fig 6.

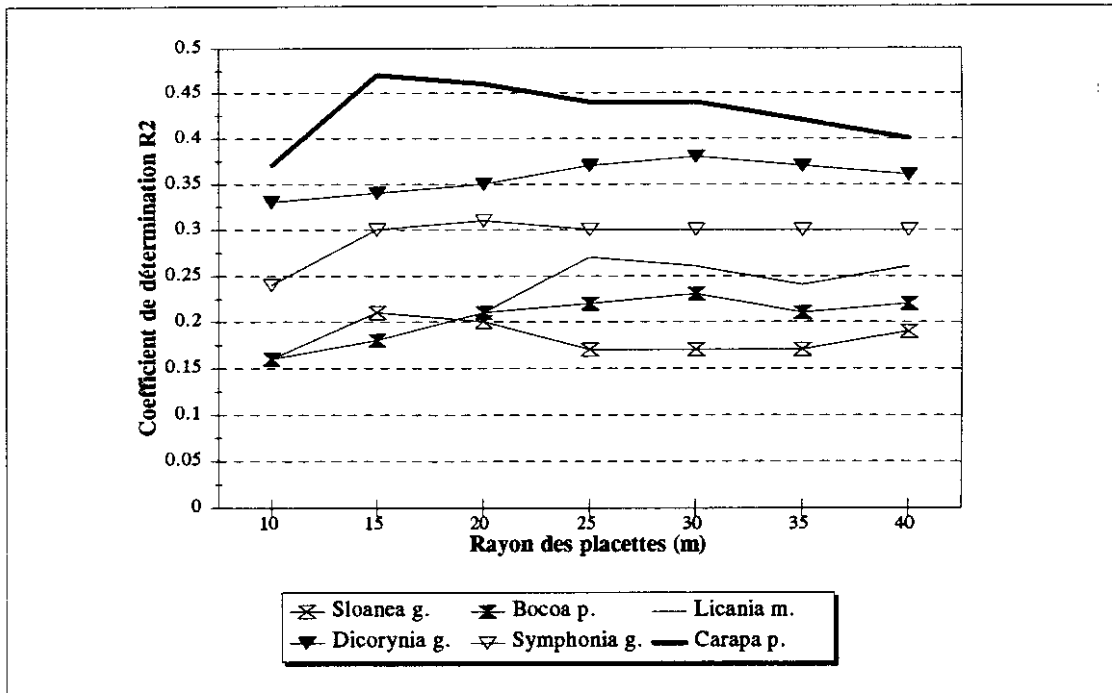


Fig 7.

ANNEXE 5

Etude de différentes formulations du modèle de croissance

SOMMAIRE

1 - Rappel des différentes formes théoriques utilisables.....	2
2 - Les différents modèles calibrés.....	2
3 - Résultats	5
3.1 - Estimation des paramètres.....	5
3.2 - Structure des résidus.....	7
3.3 - Domaine de variation	8

NB. La numérotation des équations est cohérente avec celle du texte principal.

1 - Rappel des différentes formes théoriques utilisables

Notre objectif consiste à prédire l'accroissement moyen annuel en diamètre d'un individu i à l'aide d'un modèle déterministe, a priori du type :

$$\Delta D_i = \text{potentiel}(i) * \text{réducteur}(i) \quad [2]$$

Quatre types de formulations sont utilisables (voir § 4.2.1, texte principal) :

$$\Delta D_i = \Delta D_{ipot} * \text{réducteur}(\beta_{1,i}) + \varepsilon_{1i} \quad [3]$$

avec $\Delta D_{ipot} = \text{potentiel}(\alpha_1, i)$. α_1 et β_1 sont deux ensembles de paramètres déterminés séparément sur des jeux de données différents, concernant respectivement les accroissements optimaux et tous les types d'accroissement ;

$$\Delta D_i = \text{potentiel}(\alpha_2, i) * \text{réducteur}(\beta_{2,i}) + \varepsilon_{2i} \quad [4]$$

avec α_2 et β_2 déterminés simultanément sur le même jeu de données. Les termes de « *potentiel* » et « *réducteur* » sont abusifs, il faudra plutôt parler de « *comportement moyen* » et de « *facteur modificateur* ».

$$\text{Log}\left(\frac{\Delta D_i}{\Delta D_{ipot}}\right) = \text{réducteur}(\beta_{3,i}) + \varepsilon_{3i} \quad [5]$$

avec $\Delta D_{ipot} = \text{potentiel}(\alpha_1, i)$ comme pour [3] ; α_1 et β_3 sont déterminés séparément ;

$$\text{Log}(\Delta D_i) = \text{Log}[\text{potentiel}(\alpha_4, i)] + \text{Log}[\text{réducteur}(\beta_4, i)] + \varepsilon_{4i} \quad [6]$$

avec α_4 et β_4 déterminés simultanément sur le même jeu de données (cf. [4]).

Dans les deux derniers cas, on obtient un modèle d'erreur multiplicatif. Il faut par ailleurs gérer l'occurrence des accroissements nuls et négatifs qui posent problème dans la transformation logarithmique.

2 - Les différents modèles calibrés

Nous avons choisi d'exprimer la composante « *potentiel* » (ou « *comportement moyen* », selon la formulation retenue) à l'aide d'un modèle de Korf (voir § 4.3, texte principal) :

$$\text{potentiel}(i) = \Delta D_{ipot} = amD_i [\text{Log}(K) - \text{Log}(D_i)]^{\left(1 + \frac{1}{m}\right)} \quad [7]$$

et la composante « *réducteur* » (ou « *facteur modificateur* », selon la formulation retenue) de la manière suivante (voir § 4.4.3) :

$$\text{réducteur}(i) = b\text{Log}(NBD_i + 1) + c\Delta NBD_i + d\Delta NBD_i^2 + e \quad [8]$$

ou bien :

$$\text{réducteur}(i) = e^{[b\text{Log}(NBD_i + 1) + c\Delta NBD_i + d\Delta NBD_i^2 + e]} \quad [9]$$

selon que la variable expliquée est ΔD ou sa transformée logarithmique. NBD_i et ΔNBD_i sont nos deux indices de statut social calculés sur une placette circulaire centrée sur l'arbre i et de rayon 30 m.

Ceci nous a amenés à calibrer les modèles suivants :

⇒ ajustement par étape sans transformation préalable de la variable expliquée : [3], [7] et [8] d'où

$$\Delta D_i = \left[a_1 m_1 D_i (\text{Log}(K_1) - \text{Log}(D_i)) \left(1 + \frac{1}{m_1} \right) \right] * [b_1 \text{Log}(NBD_i + 1) + c_1 \Delta NBD_i + d_1 \Delta NBD_i^2 + e_1] + \varepsilon_{1i} \quad [10]$$

Les paramètres a_1 , m_1 et K_1 sont fournis par le Tableau 4.1 et les paramètres b_1 , c_1 , d_1 par le Tableau 4.2 du texte principal (voir § 4.3 et § 4.4.3). $e_1 = 0,291$.

⇒ ajustement direct sans transformation préalable de la variable expliquée : [4], [7] et [8] d'où

$$\Delta D_i = \left[a'_2 m_2 D_i (\text{Log}(K_2) - \text{Log}(D_i)) \left(1 + \frac{1}{m_2} \right) \right] * [b_2 \text{Log}(NBD_i + 1) + c_2 \Delta NBD_i + d_2 \Delta NBD_i^2 + 1] + \varepsilon_{2i} \quad [11]$$

Même expression, mais tous les paramètres (a'_2 , m_2 , K_2 , b_2 , c_2 , d_2 , avec $a'_2 = \frac{a_2}{e_2}$) sont estimés en même temps, sur le même jeu de données. Plusieurs problèmes se sont posés avec ce modèle :

- sa formulation s'est avérée inadéquate en l'état, se traduisant par l'impossibilité de faire converger l'algorithme de minimisation des écarts. Nous avons été conduits à introduire un paramètre supplémentaire, sous forme de constante (f), de la manière suivante :

$$\Delta D_i = \left[a'_2 m_2 D_i (\text{Log}(K_2) - \text{Log}(D_i)) \left(1 + \frac{1}{m_2} \right) \right] * [b_2 \text{Log}(NBD_i + 1) + c_2 \Delta NBD_i + d_2 \Delta NBD_i^2 + 1] + f + \varepsilon_{2i} \quad [11\text{bis}]$$

L'intérêt pratique et une interprétation possible de la constante « f » sont discutés au § 4.5.2.3 (Du problème posé par les accroissements nuls et négatifs) du texte principal ;

- l'étude des résidus d'une régression non linéaire par les moindres carrés ordinaires ayant montré que la variance dépendait clairement et linéairement du diamètre des individus, nous avons cherché à utiliser des pondérations en $\frac{1}{D_i^2}$ et $\frac{1}{\sigma_j^2}$ (avec σ_j^2 variance résiduelle

¹ L'équation littérale faisant intervenir [7] et [8] est surparamétrée.

calculée par classe de diamètre j de 2 cm de large). La formulation suivante a finalement été retenue :

$$\Delta D_i = \left[a'_2 m_2 D_i (\text{Log}(K_2) - \text{Log}(D_i)) \left(1 + \frac{1}{m_2}\right) \right] * e^{[b_2 \text{Log}(NBD_i + 1) + c_2 \Delta NBD_i + d_2 \Delta NBD_i^2 + 1]} + f + \varepsilon_{2i} \quad [11\text{ter}]$$

avec $a'_2 = \frac{a_2}{e^{e_2}}$. Les paramètres ont été estimés par régression pondérée, utilisant une pondération en $\frac{1}{\sigma_j^2}$.

⇒ ajustement par étape après transformation de la variable expliquée : [5], [7] et [9] d'où

$$\text{Log} \left[\frac{\Delta D_i}{\left[a_1 m_1 D_i (\text{Log}(K_1) - \text{Log}(D_i)) \left(1 + \frac{1}{m_1}\right) \right]} + \text{cte} \right] = [b_3 \text{Log}(NBD_i + 1) + c_3 \Delta NBD_i + d_3 \Delta NBD_i^2 + e_3] + \varepsilon_{3i} \quad [12]$$

Les paramètres a_1 , m_1 et K_1 sont fournis par le Tableau 4.1 et les paramètres b_3 , c_3 , d_3 par le Tableau 4.2 du texte principal (voir § 4.3 et § 4.4.3). $e_3 = 0,401$ et $\text{cte} = 0,40^3$.

⇒ ajustement direct après transformation de la variable expliquée : [6], [7] et [9] d'où

$$\begin{aligned} \text{Log}(\Delta D_i + \text{cte}) = a'_4 + \left[\text{Log}(m_4) + \text{Log}(D_i) + \left(1 + \frac{1}{m_4}\right) \text{Log}[\text{Log}(K_4) - \text{Log}(D_i)] \right] \\ + [b_4 \text{Log}(NBD_i + 1) + c_4 \Delta NBD_i + d_4 \Delta NBD_i^2] + \varepsilon_{4i} \end{aligned} \quad [13]$$

Les paramètres a'_4 , m_4 , K_4 , b_4 , c_4 , d_4 (avec $a'_4 = \text{Log}(a_4) + e_4^4$) sont estimés en même temps, sur le même jeu de données et $\text{cte} = 0,287^5$. La variante de [13], faisant intervenir $NBD_i^{1/2}$ au lieu de $\text{Log}(NBD + 1)$, constitue la formulation [14].

Nous avons utilisé, pour ajuster les modèles [11bis], [11ter], [13] et [14], une procédure de régression non linéaire par la méthode des moindres carrés ordinaires, la transformation logarithmique ayant à peu près stabilisé la variance des observations.

A noter que le modèle [13] est le même que le modèle [11ter] à la transformation logarithmique près ; la meilleure adéquation aux données de ce dernier, comparé au modèle [11bis], montre que l'expression du *réducteur* sous la forme [9] est mieux adaptée que sous la forme [8]. Cela donne tout son intérêt au dernier modèle.

² L'équation littérale est surparamétrée.

³ Nous étudions les problèmes posés par ces constantes et la détermination de leur valeur au § 4.5.2.3 du texte principal.

⁴ L'équation littérale faisant intervenir [7] et [9] sous leur forme logarithmique est surparamétrée.

⁵ Nous étudions les problèmes posés par ces constantes et la détermination de leur valeur au § 4.5.2.3.

3 - Résultats

3.1 - Estimation des paramètres

Les modèles [11bis], [11ter] et [13] ont été ajustés aux données du jeu de calibration présenté au § 4.5.1 (voir texte principal). Ce jeu concerne 10745 individus ne présentant aucune anomalie de codage dans les circonférences pendant la période 1988-1991, vivants au moins depuis 1985 et situés à plus de 30 m des bords des parcelles (ceci pour permettre le calcul des indices *NBD* et ΔNBD). Les Palmiers et les arbres blessés et/ou dévitalisés ayant survécu aux traitements sont éliminés. Les principaux résultats des ajustements sont donnés dans le Tableau 1 ci-dessous.

Tableau 1. Paramètres des équations [11bis], [11ter] et [13], estimés à partir des données portant sur 10745 arbres, toutes espèces confondues (sauf Palmiers).

Paramètre	Pas de transformation de la variable dépendante			Transformation logarithmique de la variable dépendante	
	Pour mémoire : équation [10] (ajustement par étape)	Equation [11bis] (ajustement direct)	Equation [11ter] (ajustement direct, réducteur modifié) ^e	Pour mémoire : équation [12] (ajustement par étape)	Equation [13] (ajustement direct) ^f
	Estimation (σ)	Estimation (σ)	Estimation (σ)	Estimation (σ)	Estimation (σ)
a (année ⁻¹)	0,025 (0,007)	-	-	0,025 (0,007)	-
a' (a/e) (*10 ⁻³)	-	1,84 (0,65)	-	-	-
a' (a/e ^e) (*10 ⁻³)	-	-	1,91 (0,30)	-	-
a' (Log(a)+e)	-	-	-	-	-5,21 (0,11)
m	0,602 (0,116)	0,694 (0,128)	0,586 (0,106)	0,602 (0,116)	1,088 (0,150)
K (cm)	180 (31)	647 (218)	341 (76)	180 (31)	208 (25)
b (*10 ⁻³)	-59,46 (1,12)	-68,93 (21,57)	-164,09 (17,37)	-101,05 (1,88)	-114,08 (14,08)
c (*10 ⁻³)	-3,69 (0,15)	-21,15 (5,82)	-15,00 (1,37)	-7,23 (0,26)	-18,32 (0,59)
d (*10 ⁻³)	-0,02 (0,00)	-0,12 (0,03)	-0,10 (0,01)	-0,05 (0,00)	-0,13 (0,01)
e	0,291 (0,01)	-	-	-0,401 (0,01)	-
f	-	-0,298 (0,059)	-0,405 (0,117)	-	-
SCR ^a	106	458	10807 (pond.)	297	1320
SCT ^b	138	587	14117 (pond.)	395	1715
σ^2 ^c	0,010 (cm ²)	0,043 (cm ²)	0,043 (cm ²)	0,028 (Log(cm) ²)	0,123 (Log(cm) ²)
(Pseudo-)R ² ^d	0,234	0,221	0,234	0,249	0,230

^a Somme des carrés des résidus.

^b $SCT = \sum (y_i - \bar{y})^2$

^c Carré moyen des résidus (estimation de la variance résiduelle) : SCR/ddl.

^d $1 - (SCR/SCT)$.

^e Régression non linéaire utilisant une pondération en $1/\sigma_i^2$.

^f Si on utilise la variable $NBD^{1/2}$ en lieu et place de $\text{Log}(NBD+1)$ (équation [14]), les valeurs des paramètres sont estimées comme suit : a' = -6,77 ($\sigma = 0,67$), m = 0,514 ($\sigma = 0,119$), K = 693 ($\sigma = 298$), b = -31,71*10⁻³ ($\sigma = 4,89*10^{-3}$), c = -18,41*10⁻³ ($\sigma = 0,59*10^{-3}$), d = -0,13*10⁻³ ($\sigma = 0,01*10^{-3}$). SCR = 1323 et $\sigma^2 = 0,123$ (Log(cm)²).

Les valeurs estimées sont toutes significativement non nulles. Nous nous sommes assurés de l'intérêt de l'inclusion des variables $\text{Log}(NBD + 1)$ et ΔNBD^2 lorsque ΔNBD est déjà pris en compte, en réalisant des tests F sur des modèles emboîtés. Ainsi par exemple, pour le modèle [13], l'ajout de $\text{Log}(NBD + 1)$ à une partie « réducteur » ne comprenant que ΔNBD amène $F=92,57$ (resp. $F=65,64$ lorsque le « réducteur » fait déjà intervenir ΔNBD et ΔNBD^2) et $F=179,85$ lorsque l'on complète avec la variable ΔNBD^2 (valeurs à comparer avec $F_{0,01}(1, \infty)=6,63$). Les mêmes tests peuvent être réalisés avec la variable $NBD^{1/2}$ (modèle [14], aboutissant aux mêmes conclusions.

On note que le modèle [13] est celui qui permet l'estimation la plus précise du paramètre K. Pour les autres paramètres, ce sont les modèles ajustés par étape ([10] et [12]) qui de ce point de vue donnent les meilleurs résultats ; la variance résiduelle est plus faible qu'avec les modèles équivalents ajustés directement (resp. [11bis] et [13]). Dans les régressions directes, c'est le modèle [13] qui se comporte le mieux, pour tous les paramètres, excepté m.

3.2 - Structure des résidus

Aucune des distributions résiduelles n'est normale (voir Tableau 2). Il existe cependant des différences notables entre les cinq modèles, comme illustré par la Fig.1 ci-dessous.

Tableau 2. Caractéristiques principales des distributions résiduelles issues de l'ajustement des cinq modèles (10745 arbres pris en compte).

Caractéristiques de la distribution	Pas de transformation de la variable dépendante			Transformation logarithmique de la variable dépendante	
	Modèle [10] (ajustement par étape)	Modèle [11bis] (ajustement direct)	Modèle [11ter] (ajustement direct, réducteur modifié)	Modèle [12] (ajustement par étape)	Modèle [13] ^c (ajustement direct)
Dissymétrie ^a (g1)	1,87	1,84	1,88	0,73	0,17
Applatissement ^a (g2)	8,95	8,78	8,88	4,67	0,60
D ^b	0,097 (**)	0,098 (**)	0,104 (**)	0,070 (**)	0,025 (**)

^a Coefficients de Fisher, nuls lorsque la distribution est normale.

^b Statistique de Kolmogorov. L'hypothèse de normalité est rejetée, avec un risque d'erreur inférieur à 1%, lorsque $D \geq 1,63/\sqrt{n}$ (Dagnelie, 1975) soit dans notre cas 0,016.

^c Dans le cas du modèle [14] : $g1 = 0,17$, $g2 = 0,62$ et $D = 0,024$.

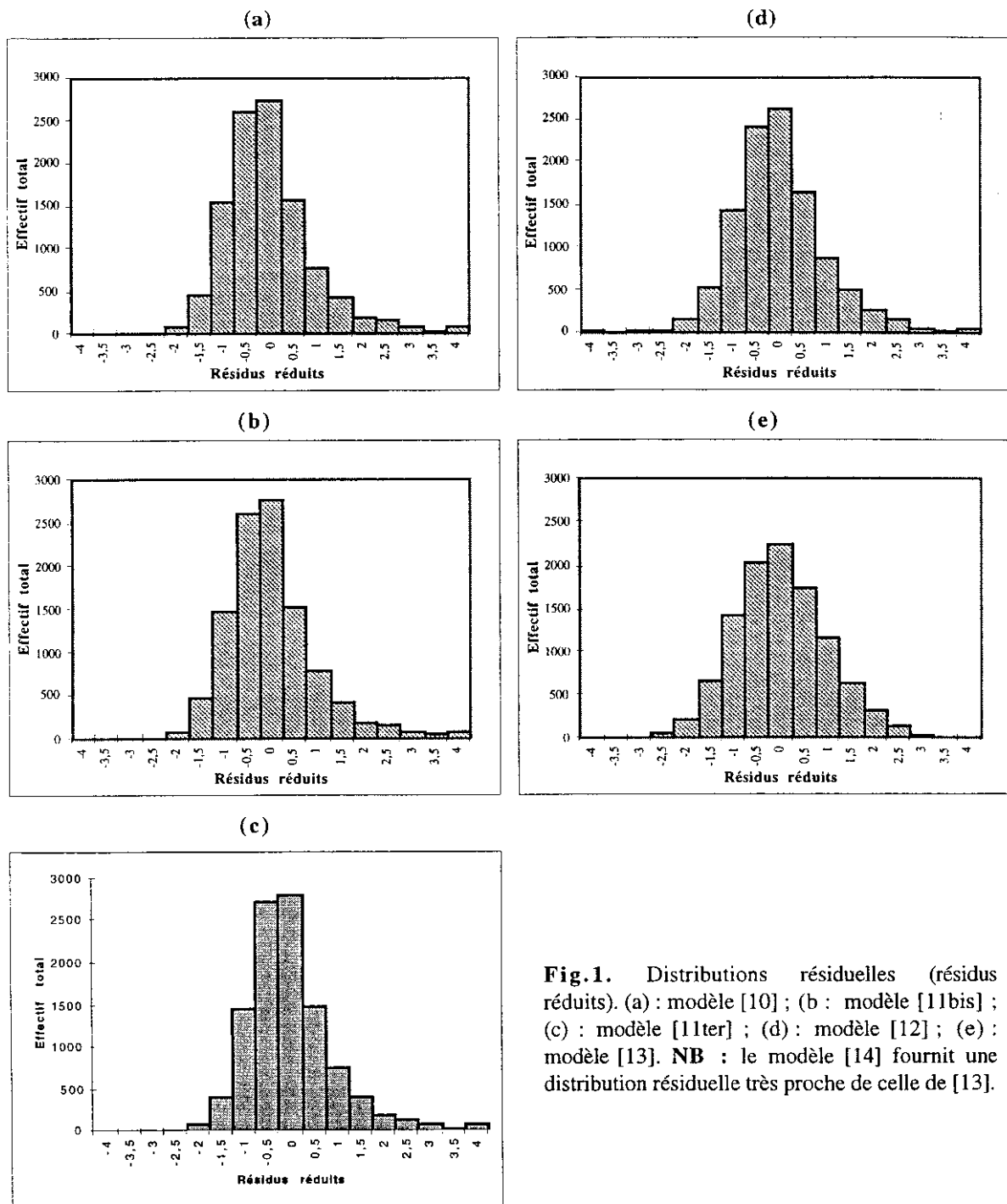


Fig.1. Distributions résiduelles (résidus réduits). (a) : modèle [10] ; (b : modèle [11bis] ; (c) : modèle [11ter] ; (d) : modèle [12] ; (e) : modèle [13]. NB : le modèle [14] fournit une distribution résiduelle très proche de celle de [13].

Les trois modèles ajustés sans transformation de la variable dépendante fournissent des résidus d'allure log-normale, avec des coefficients de Fisher positifs et relativement importants en valeur absolue. L'utilisation de la transformation logarithmique améliore nettement les choses, comme on pouvait s'y attendre étant donnée la distribution observée sur les accroissements (cf. Fig4.1, texte principal).

Le modèle [13] fournit, de loin, les résidus les mieux distribués ce qui constitue un avantage pour la réalisation de tests statistiques ultérieurs. Leur représentation en fonction des variables explicatives (voir Fig.a13 en annexe 6) ne laisse par ailleurs apparaître aucune structure particulière.

3.3 - Domaine de variation

Nous avons représenté ci-dessous le domaine de variation des modèles [11bis], [11ter] et [13]. Dans ces modèles, la composante « *réducteur* » varie dans un intervalle $[B_{inf}, B_{sup}]$ avec $B_{inf} < 1$ et $B_{sup} > 1$. La composante « *potentiel* » ne représente plus l'accroissement potentiel maximal des individus, mais un accroissement moyen dans un environnement « neutre » c'est-à-dire qui ne freine pas la croissance (cas où *réducteur* < 1) mais qui ne la stimule pas non plus (cas où *réducteur* > 1). En conséquence, il est plus correct de parler respectivement de « *comportement moyen* » (CM) et de « *facteur modificateur* » (FM) en lieu et place de « *potentiel* » et « *réducteur* ».

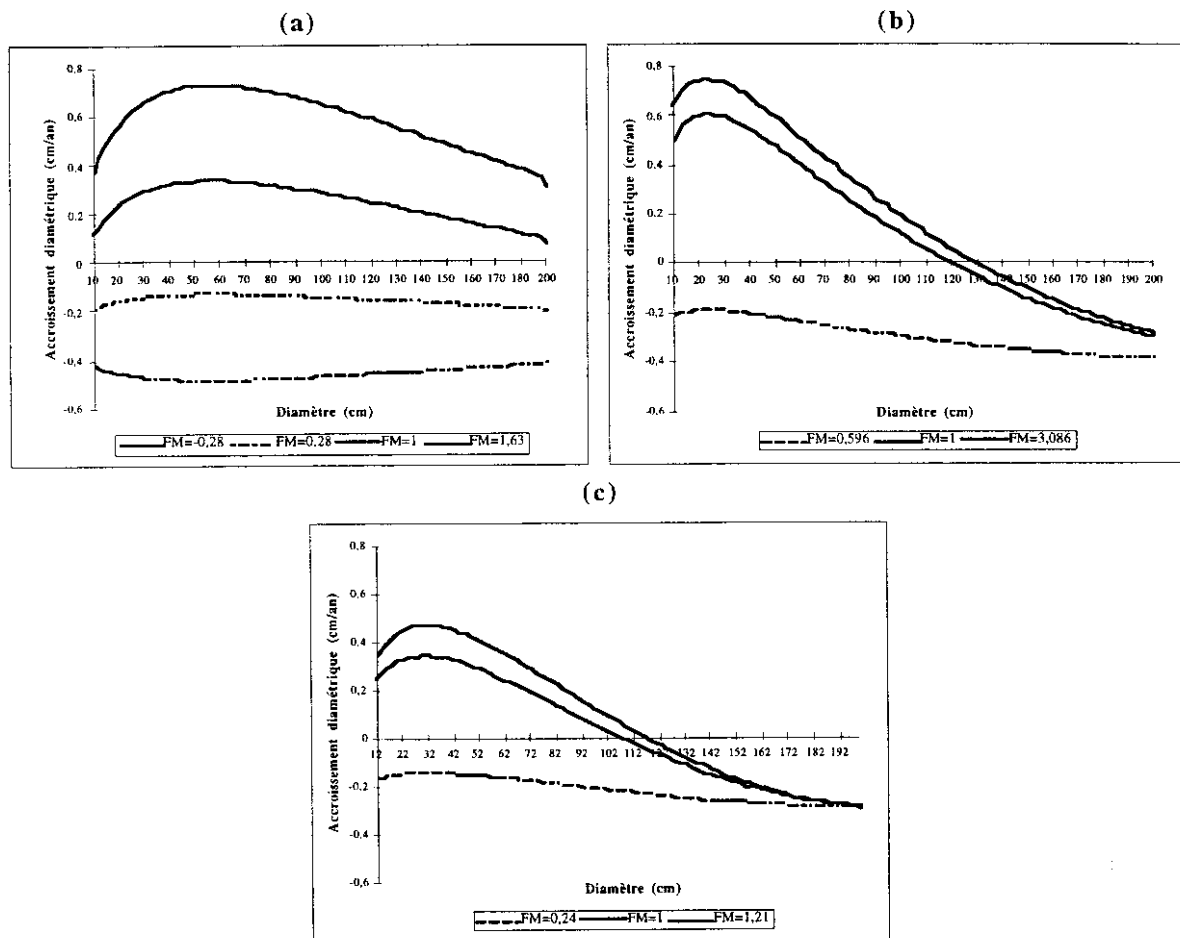


Fig.2. Domaine de variation du modèle de croissance en fonction du diamètre, lorsque l'ensemble des paramètres est ajusté directement sur le jeu de données ($FM = \text{facteur modificateur}$). (a) Formulation [11bis]. La deuxième plus petite valeur de FM vaut $0,28^6$. (b) Formulation [11ter]. (c) Formulation [13]. **NB1** : dans ces représentations, le paramètre a du modèle de Korf agglomère toutes les constantes initialement présentes dans le facteur modificateur : pour [11bis], on travaille avec $a'=a/e$, [11ter] $a'=a/(e^{e+1})$ et [13] : $a'=\text{Log}(a)+e$. Cela revient à rendre le comportement moyen optimal. **NB2** : aucun individu ne dépasse 101 cm de diamètre dans le jeu de données.

⁶ La valeur $FM=-0,28$ correspond au cas extrême d'un individu dont le nombre de voisins « dominants » est passé de 139 à 175 ($\Delta NBD=36$). La plus petite valeur suivante vaut 0,28 ($NBD=155$ et $\Delta NBD=16$).

ANNEXE 6

Figures et Tableaux

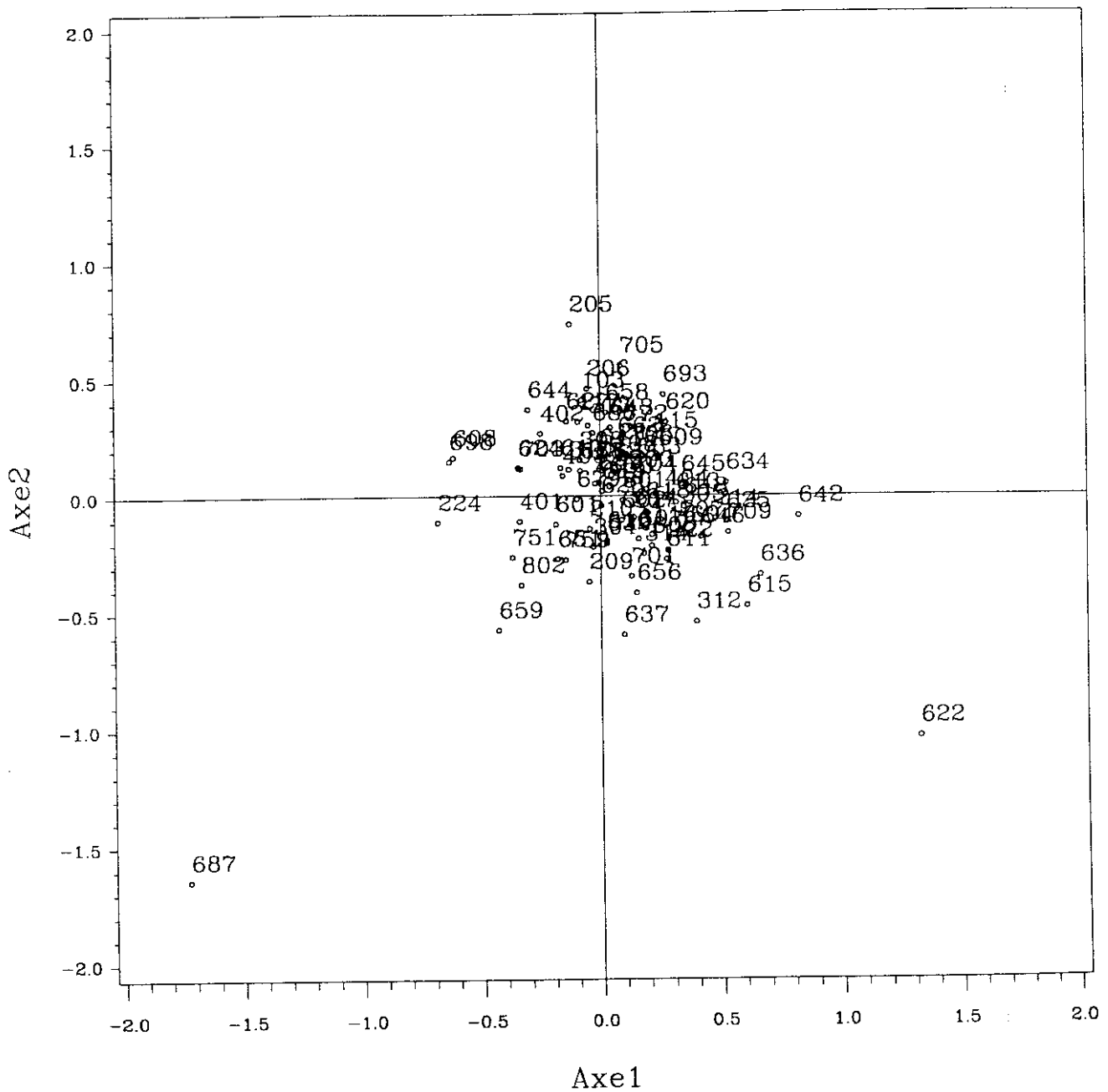
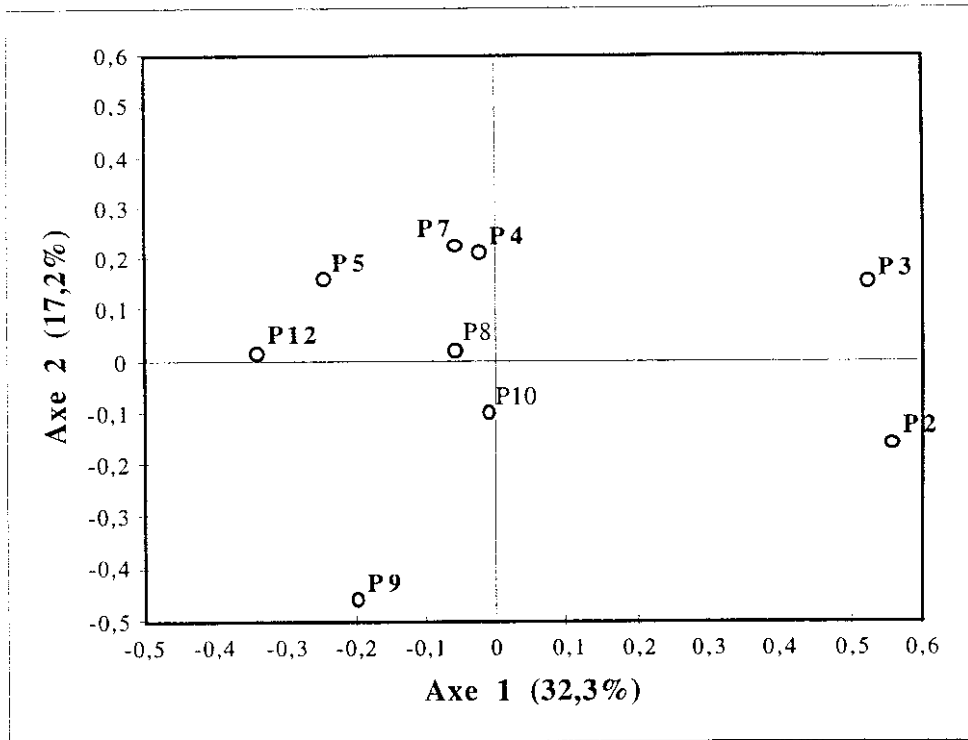


Fig.a1. Projection des 87 EGE sur le premier plan factoriel de l'AFC « espèce*parcelle » présentée au paragraphe 1.2.3.2 (voir texte principal) : la correspondance avec les noms d'espèces figure plus loin, dans le Tableau a8. Les individus pris en compte sont ceux des jeux de données utilisés pour l'étude de la croissance (cf. chapitre 4).

(a)



(b)

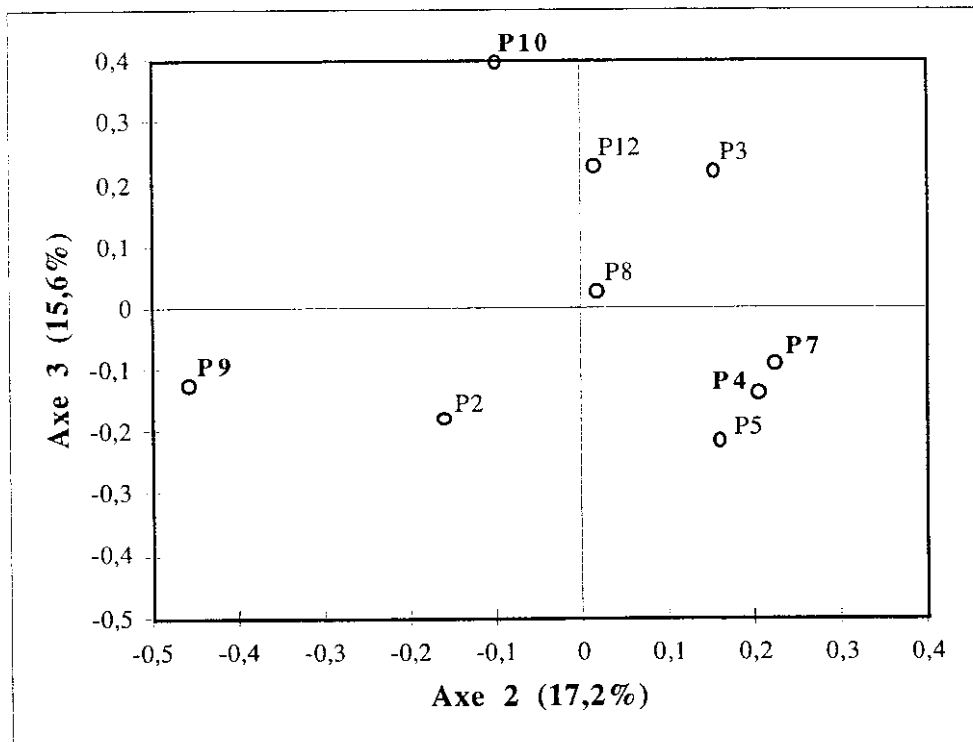


Fig.a2. Projection des points « parcelle » sur (a) le premier plan factoriel, et (b) le deuxième plan factoriel de l'AFC espèce*parcelle, lorsque les parcelles témoins sont éliminées. Les points en « gras » sont les mieux représentés sur le plan (*ie* le \cos^2 de leur angle avec l'un ou l'autre des axes est supérieur à 0,20). La table de contingence initiale recense les individus utilisés dans les jeux de données du chapitre 4 appartenant aux 87 espèces les mieux représentées (voir texte). Il s'agit d'effectifs présents en 1988 (campagne 5), après les traitements sylvicoles.

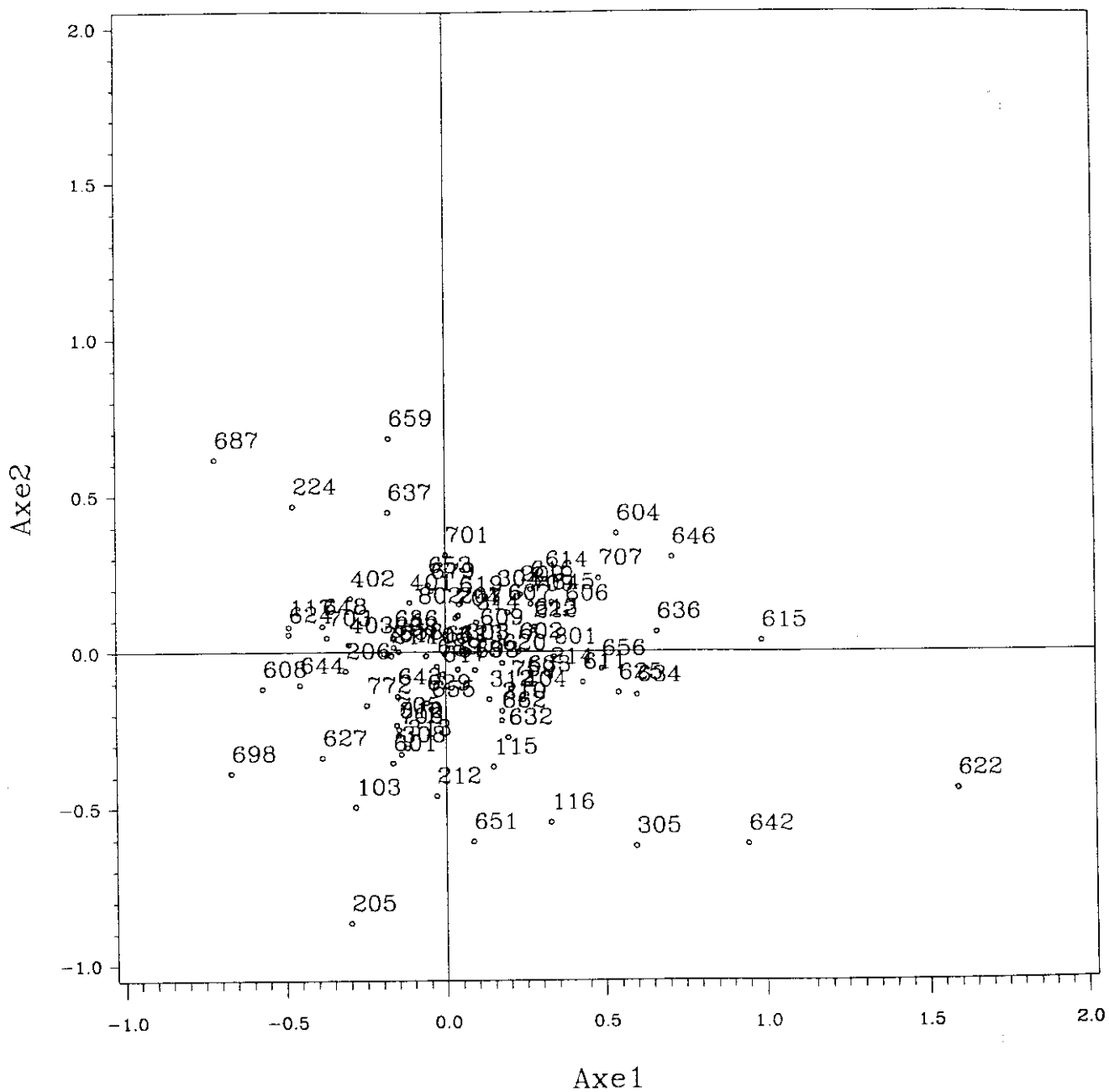


Fig.a3 (a). Projection des 87 EGE sur le premier plan factoriel de l'AFC « espèce*parcelle » présentée au paragraphe 1.2.3.2 (voir texte principal), lorsque les 3 parcelles témoins 1, 6 et 11 sont éliminées. La correspondance avec les noms d'espèces figure plus loin, dans le Tableau a8. Les individus pris en compte sont ceux des jeux de données utilisés pour l'étude de la croissance (cf. chapitre 4).

Voir **Fig.a3 (b)** page suivante.

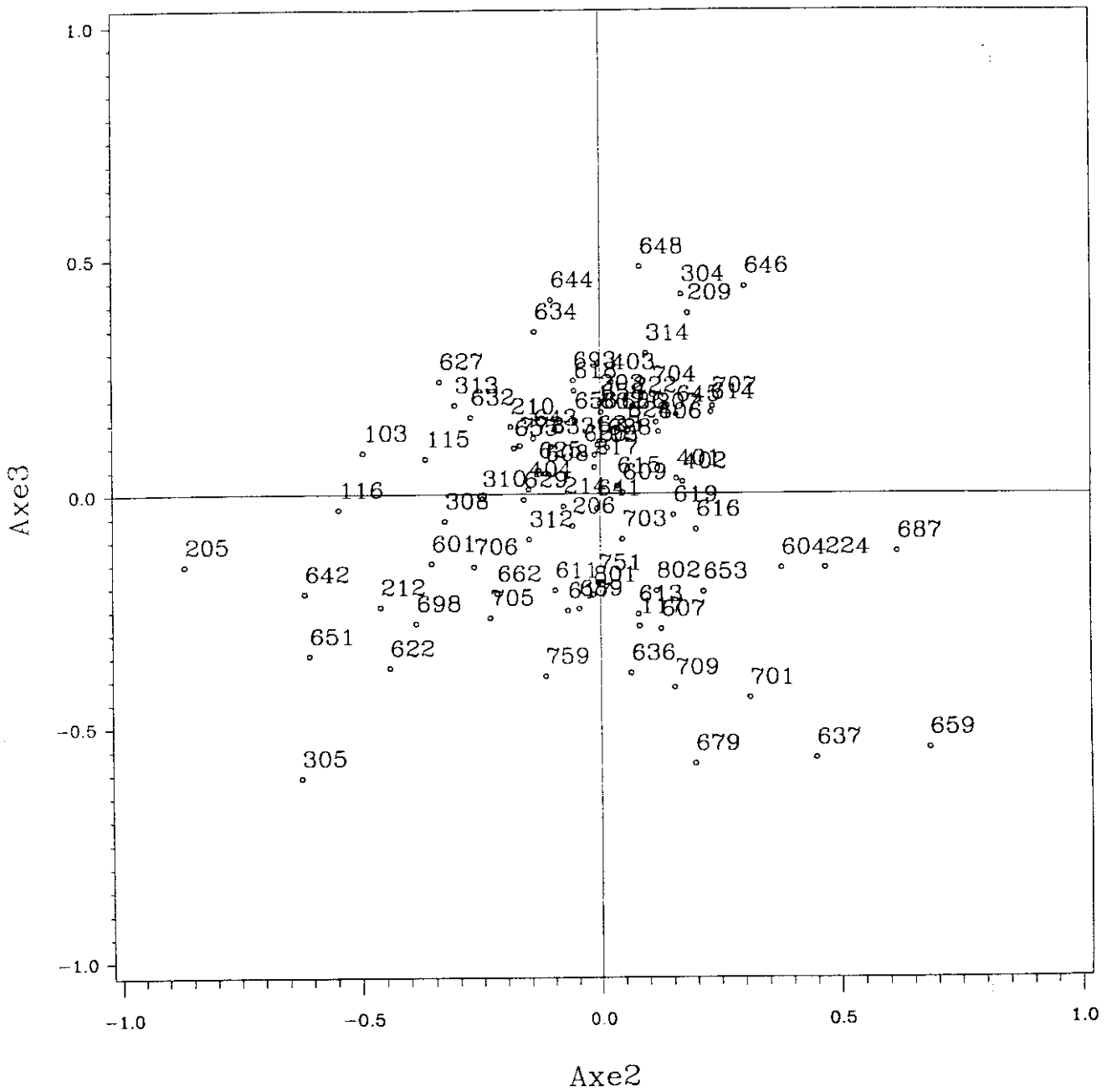


Fig.a3 (b). Projection des 87 EGE sur le deuxième plan factoriel de l'AFC « espèce*parcelle » présentée au paragraphe 1.2.3.2 (voir texte principal), lorsque les 3 parcelles témoins 1, 6 et 11 sont éliminées. La correspondance avec les noms d'espèces figure plus loin, dans le Tableau a8.

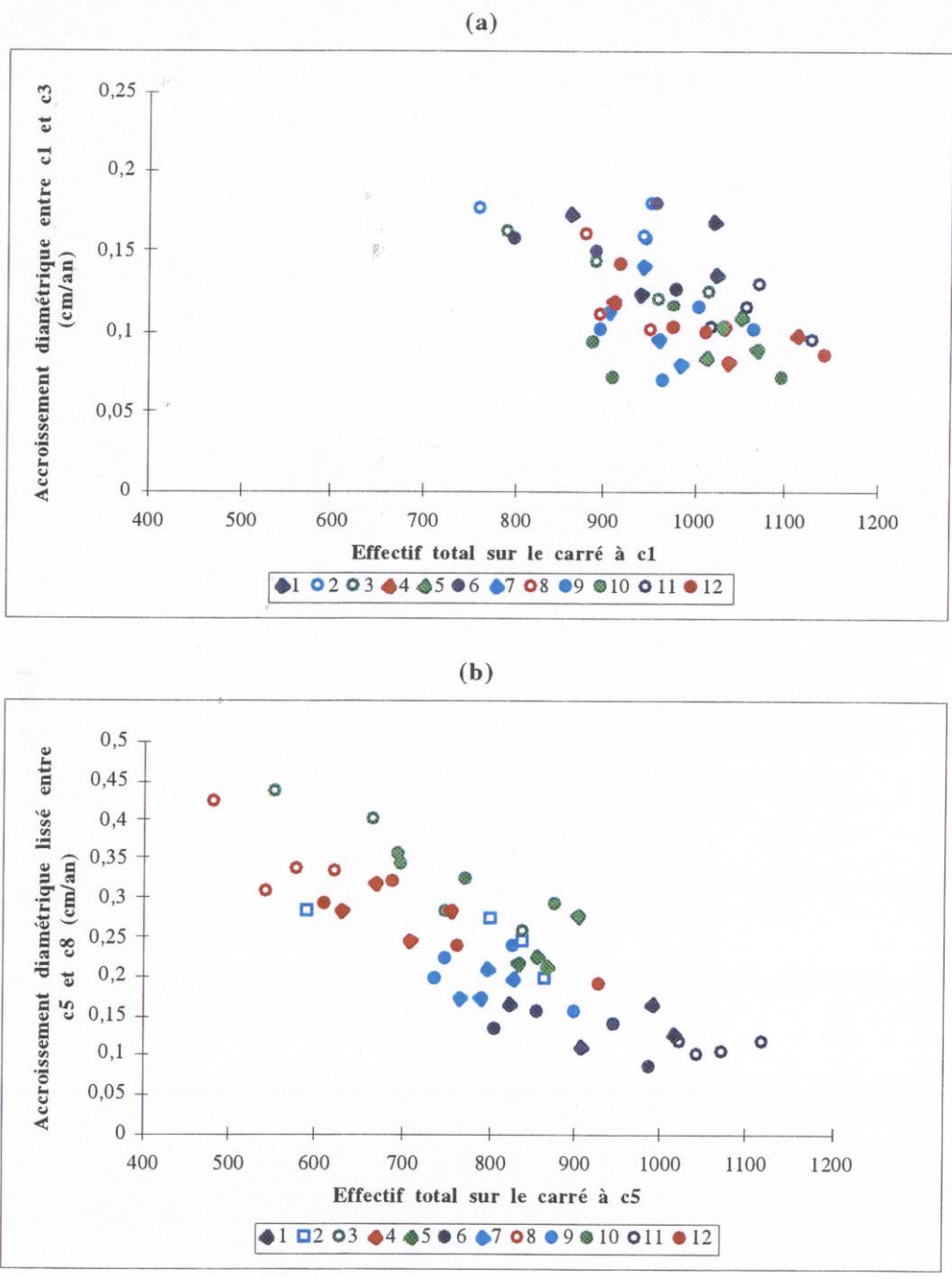


Fig.a4. Accroissement moyen en diamètre sur les différents carrés de chaque parcelle, en fonction de l'effectif présent : (a) avant traitement (entre les campagnes 1 et 3) et (b) après traitement. Dans ce cas, les accroissements ont été lissés sur trois ans, entre la campagne 5 et la campagne 8, selon la méthode exposée au paragraphe 4.1. Les quatre couleurs correspondent aux quatre traitements appliqués et les figurés en plein indiquent les parcelles qui serviront à ajuster le modèle de croissance (voir chapitre 4).

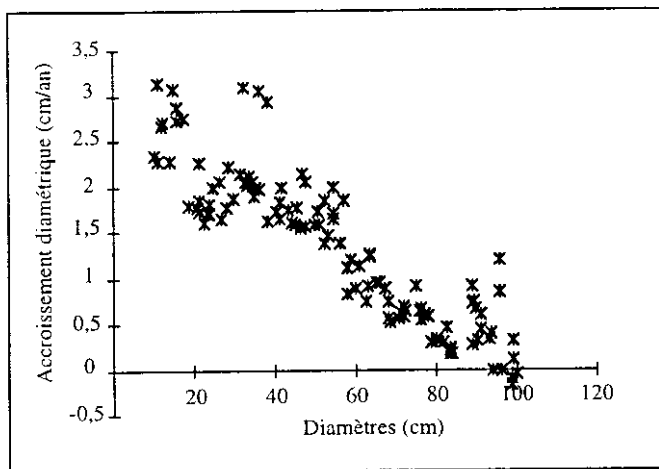


Fig.a5. Représentation des accroissements les plus importants, après lissage sur une période de 3 ans (1984 à 1990, sur 8 parcelles).

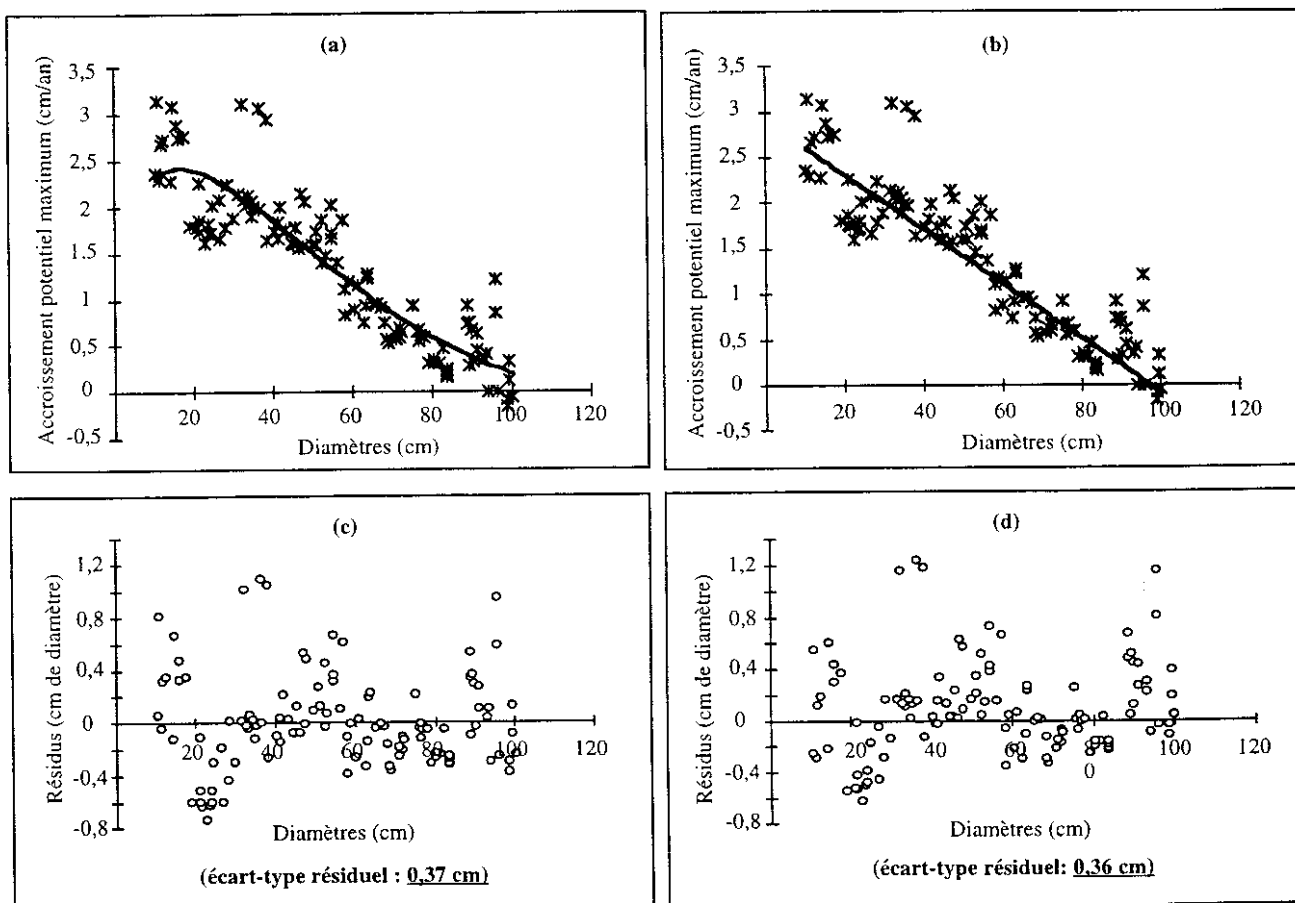


Fig. a6. (a) Ajustement par un modèle de Korf. (b) Ajustement par une droite. (c) et (d) Résidus respectifs des ajustements.

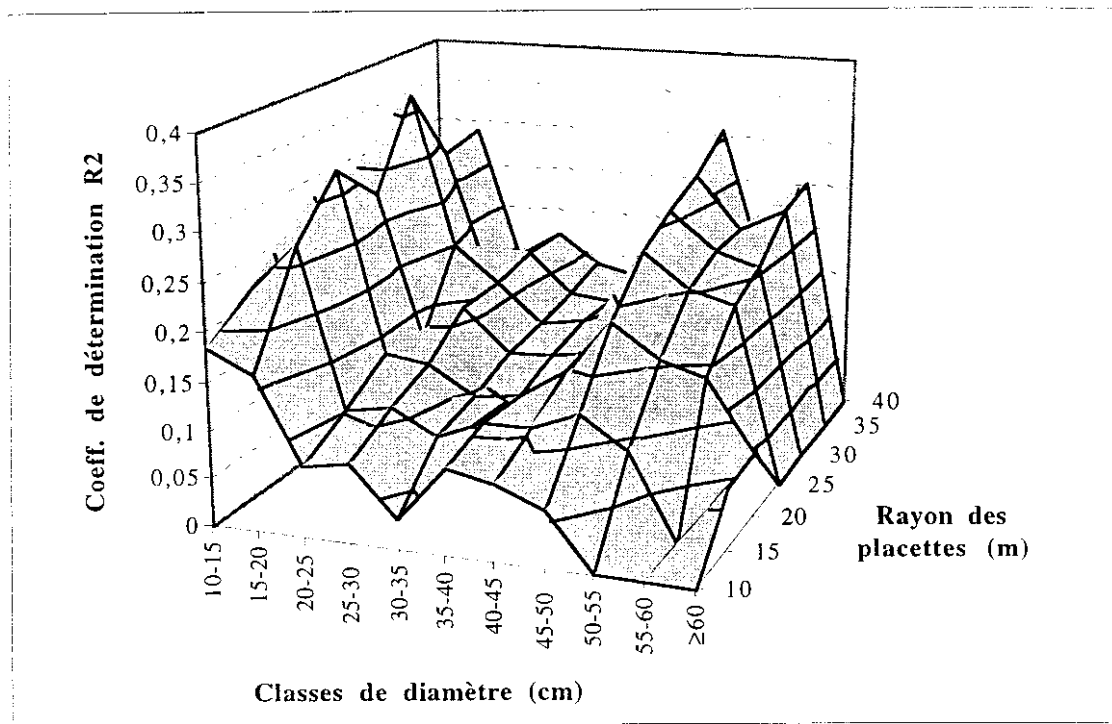


Fig.a6bis. Comparaison toutes espèces confondues, par classe de diamètre et pour différents rayons de placette de calcul, des coefficients de détermination R^2 obtenus avec le modèle linéaire $\text{Log}(\Delta D_i + cte_i) = b \cdot IC1_i + c \cdot IC2_i + d$; $IC1$ (et $IC2$) : indice(s) qui maximise(nt) la valeur du R^2 dans chaque cas.

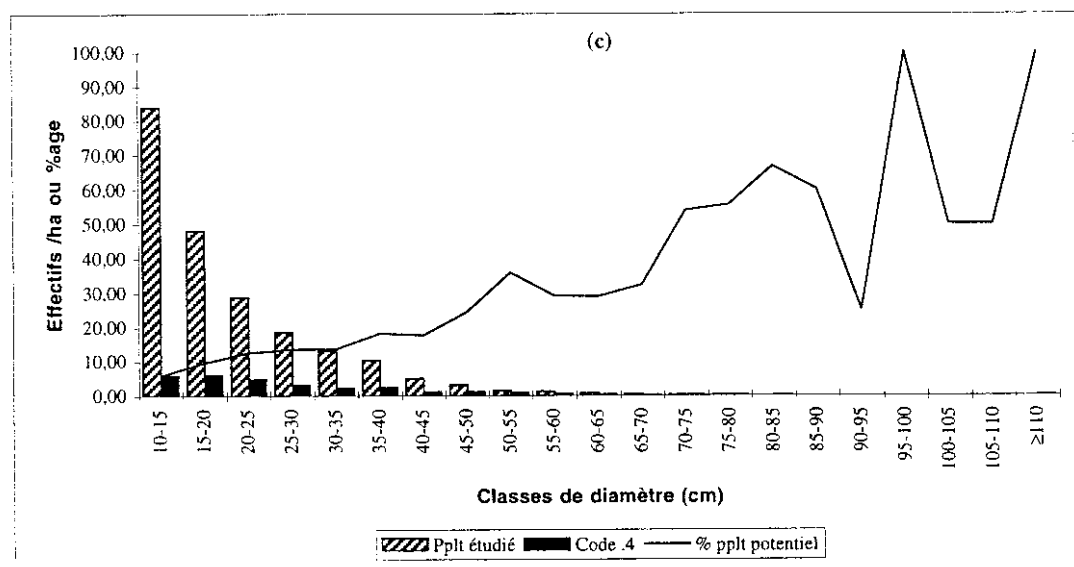
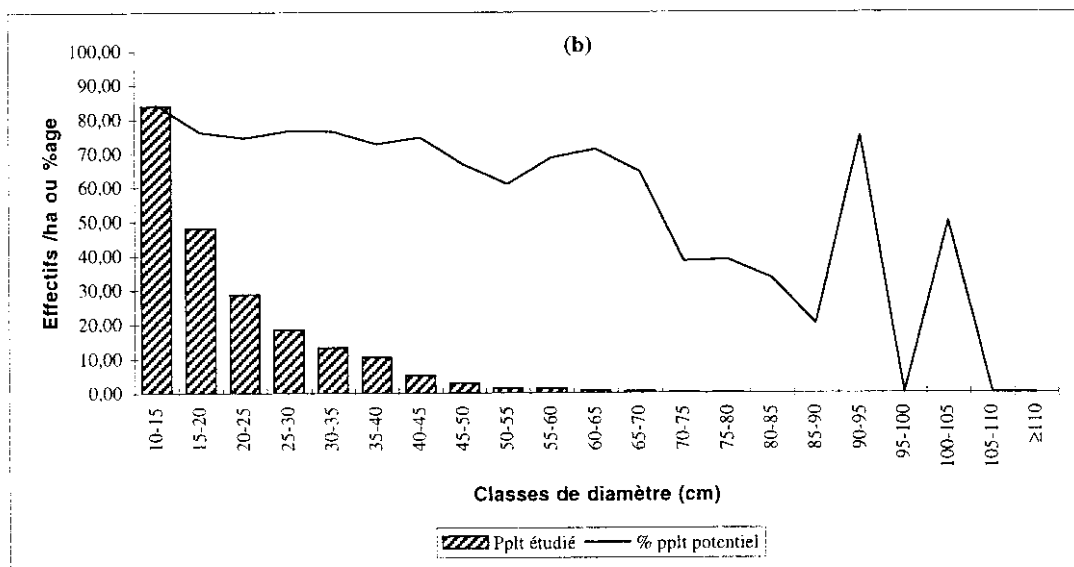
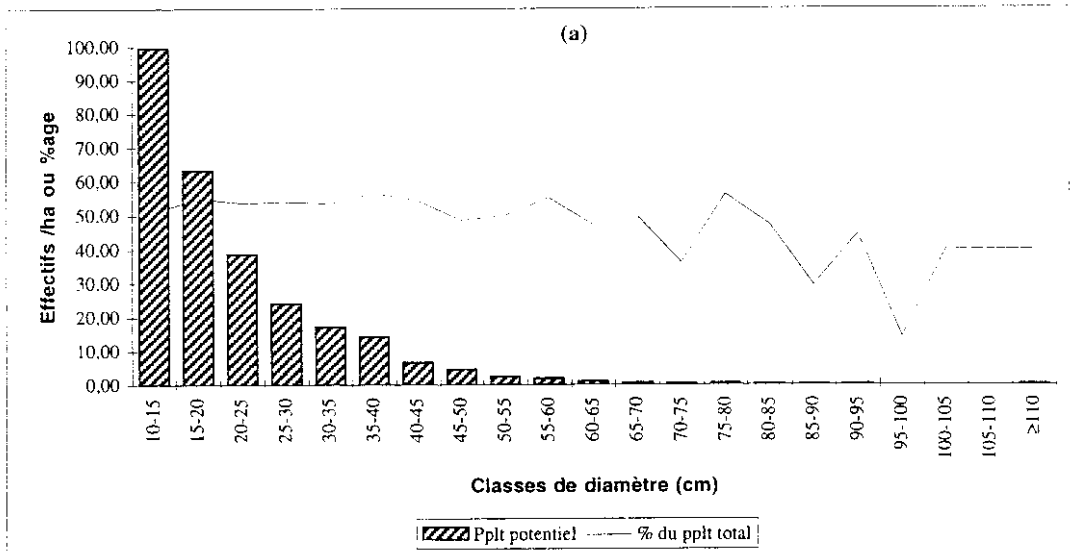


Fig.a7. Structure, en 1988, (a) du peuplement potentiellement étudiable, constitué des arbres vivants entre 1985 et 1991 et situés à plus de 30 m des bords des huit parcelles d'étude ; (b) du peuplement réellement étudié (10 745 arbres) ; (c) de la fraction du peuplement potentiellement étudiable constituée des arbres de section non circulaire à 1m30.

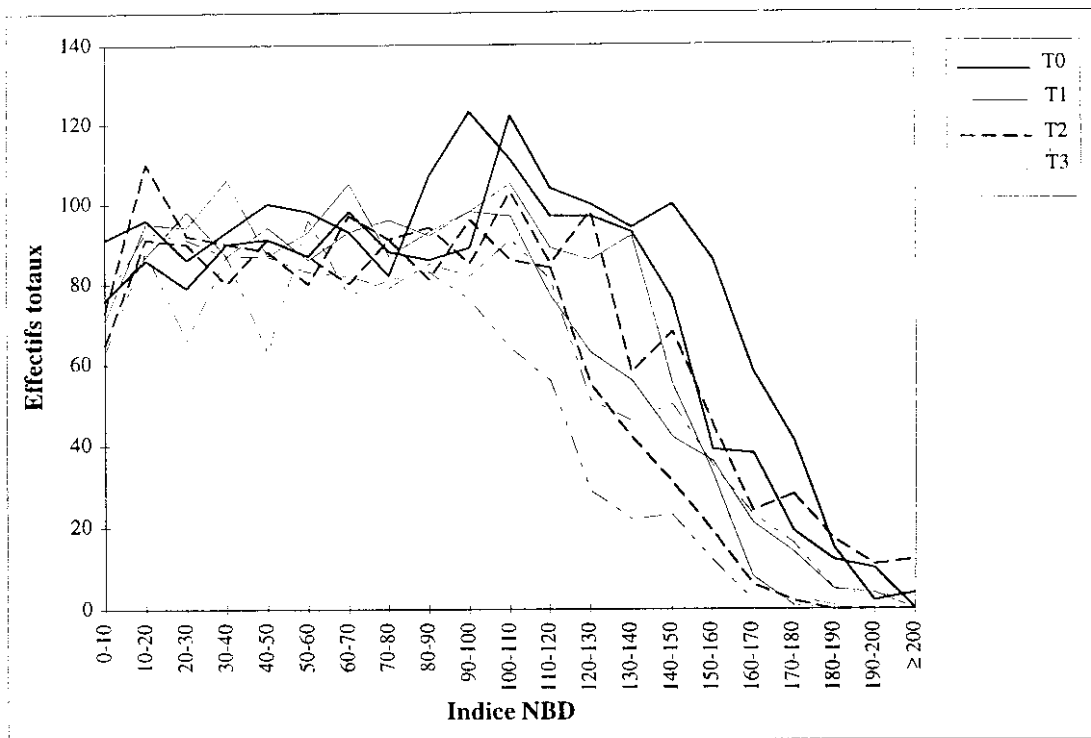


Fig.a8. Distribution de l'indice NBD, calculé en 1988 (10745 individus pris en compte), en fonction des parcelles et des traitements sylvicoles. Chaque figuré représente un traitement. Chaque courbe représente une parcelle.

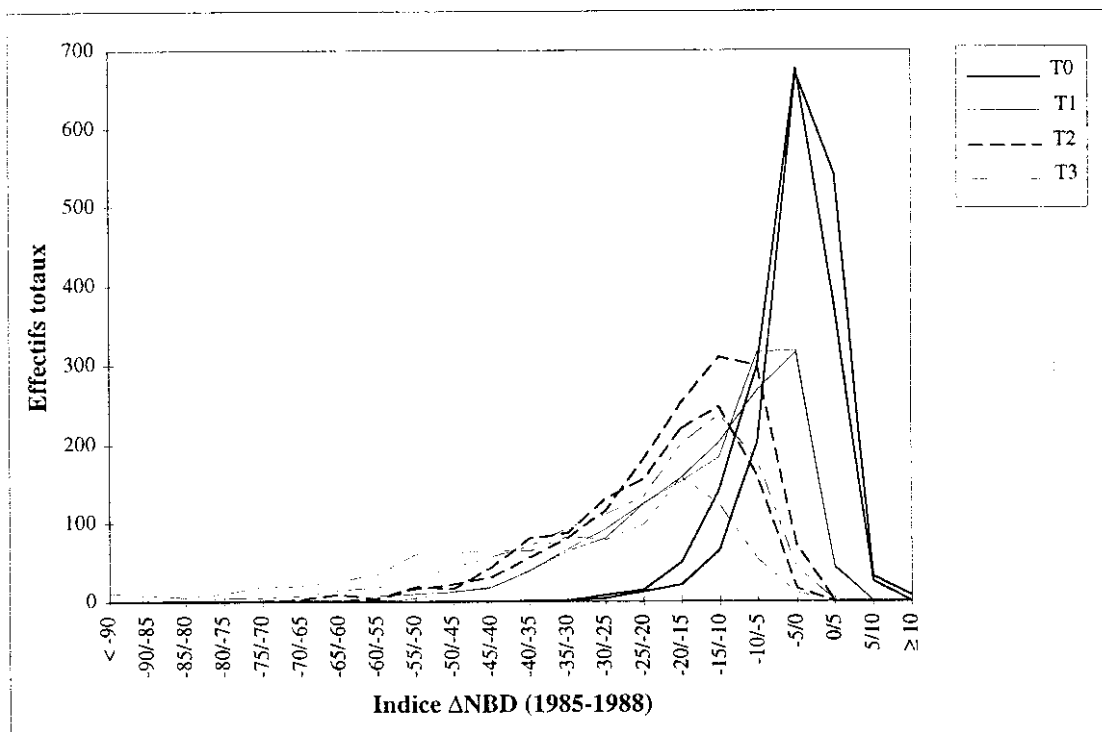
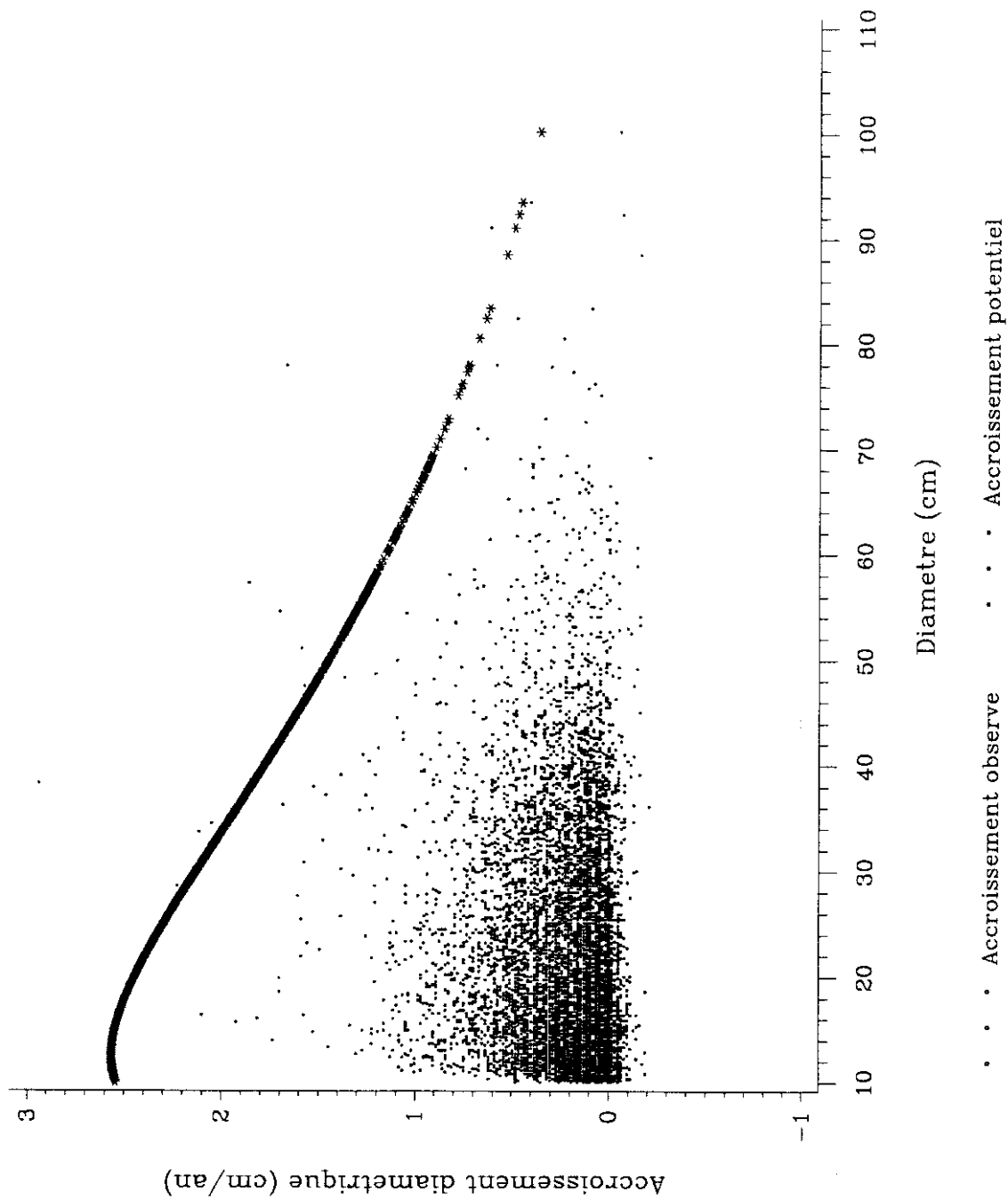


Fig.a9. Distribution de l'indice Δ NBD, calculé sur la période 1985-1988 (10745 individus pris en compte), en fonction des parcelles et des traitements sylvicoles. Chaque figuré représente un traitement. Chaque courbe représente une parcelle.

Fig. a 10. Accroissements moyens annuels lissés sur trois ans des 10745 arbres sur trois ans des données de calibration, sur lesquels on a superposé les valeurs prédites du « *potentiel* » calculées à l'aide du modèle [7] (paramètres fournis au Tableau 4.1, avec $K=180$ cm).



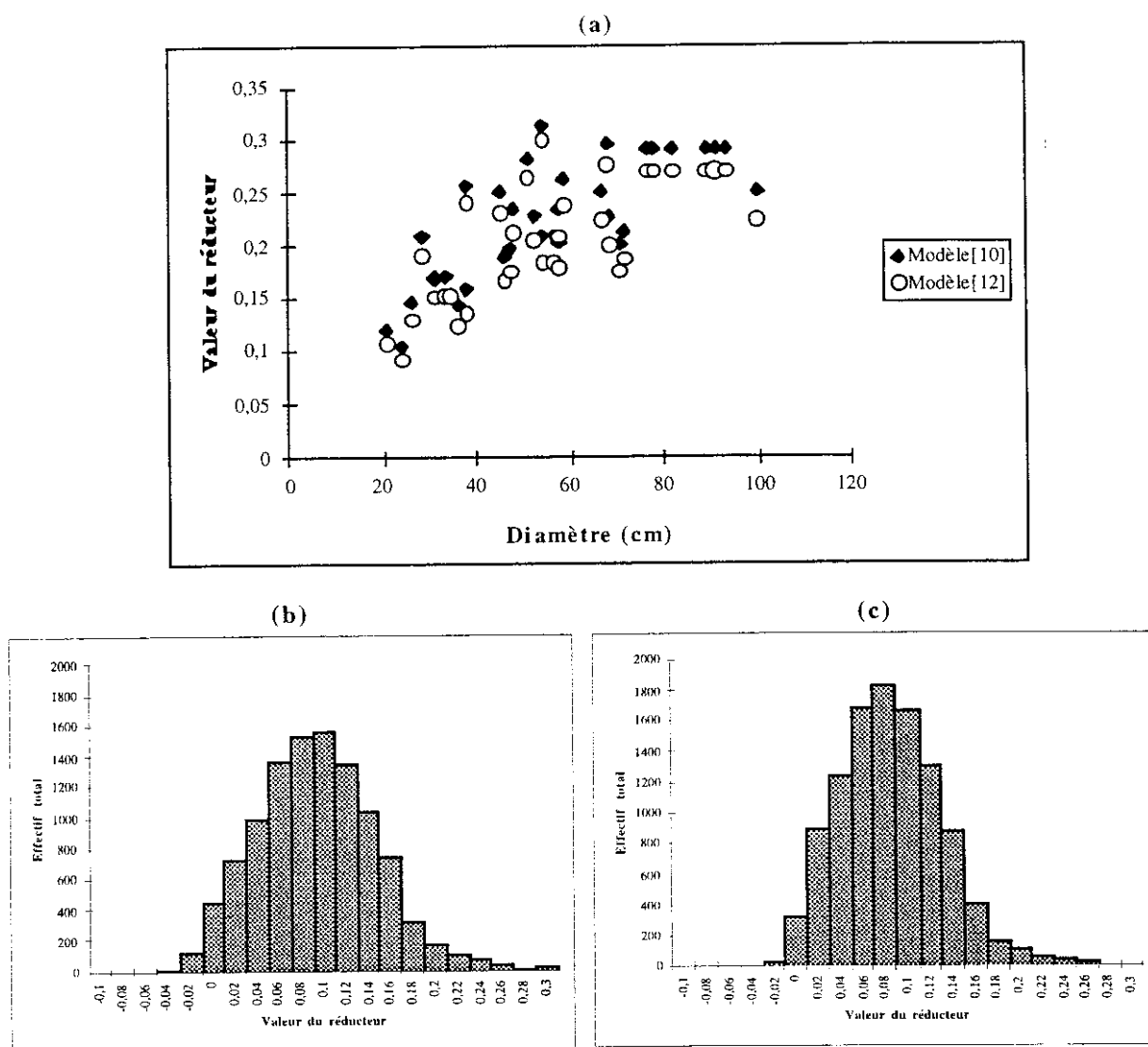


Fig.a11. (a) Valeur du « réducteur » pour quelques arbres présentant des accroissements maxima (35 cas étudiés). (b) Distribution des valeurs de la composante « réducteur » pour le modèle [10] et (c) pour le modèle [12] (10745 arbres concernés). Les faibles valeurs observées s'expliquent par la position du nuage des points (D_i , ΔD_i) par rapport à la composante « potentiel » : voir figure a10 à la page précédente.

Rappel :

Modèle [10] :

$$\Delta D_i = \left[a_1 m_1 D_i (\text{Log}(K_1) - \text{Log}(D_i)) \left(1 + \frac{1}{m_1} \right) \right] * [b_1 \text{Log}(NBD_i + 1) + c_1 \Delta NBD_i + d_1 \Delta NBD_i^2 + e_1] + \varepsilon_{1i}$$

Modèle [12] :

$$\text{Log} \left(\frac{\Delta D_i}{\left[a_1 m_1 D_i (\text{Log}(K_1) - \text{Log}(D_i)) \left(1 + \frac{1}{m_1} \right) \right]} + cte \right) = [b_3 \text{Log}(NBD_i + 1) + c_3 \Delta NBD_i + d_3 \Delta NBD_i^2 + e_3] + \varepsilon_{3i}$$

Les valeurs des paramètres figurent dans les Tableaux 4.1 et 4.2 du texte principal.

Fig.a12(a). Représentation des accroissements moyens annuels observés sur les 10745 individus du fichier de calibration en fonction du diamètre à 1m30, avec superposition des valeurs prédites par le modèle [13].

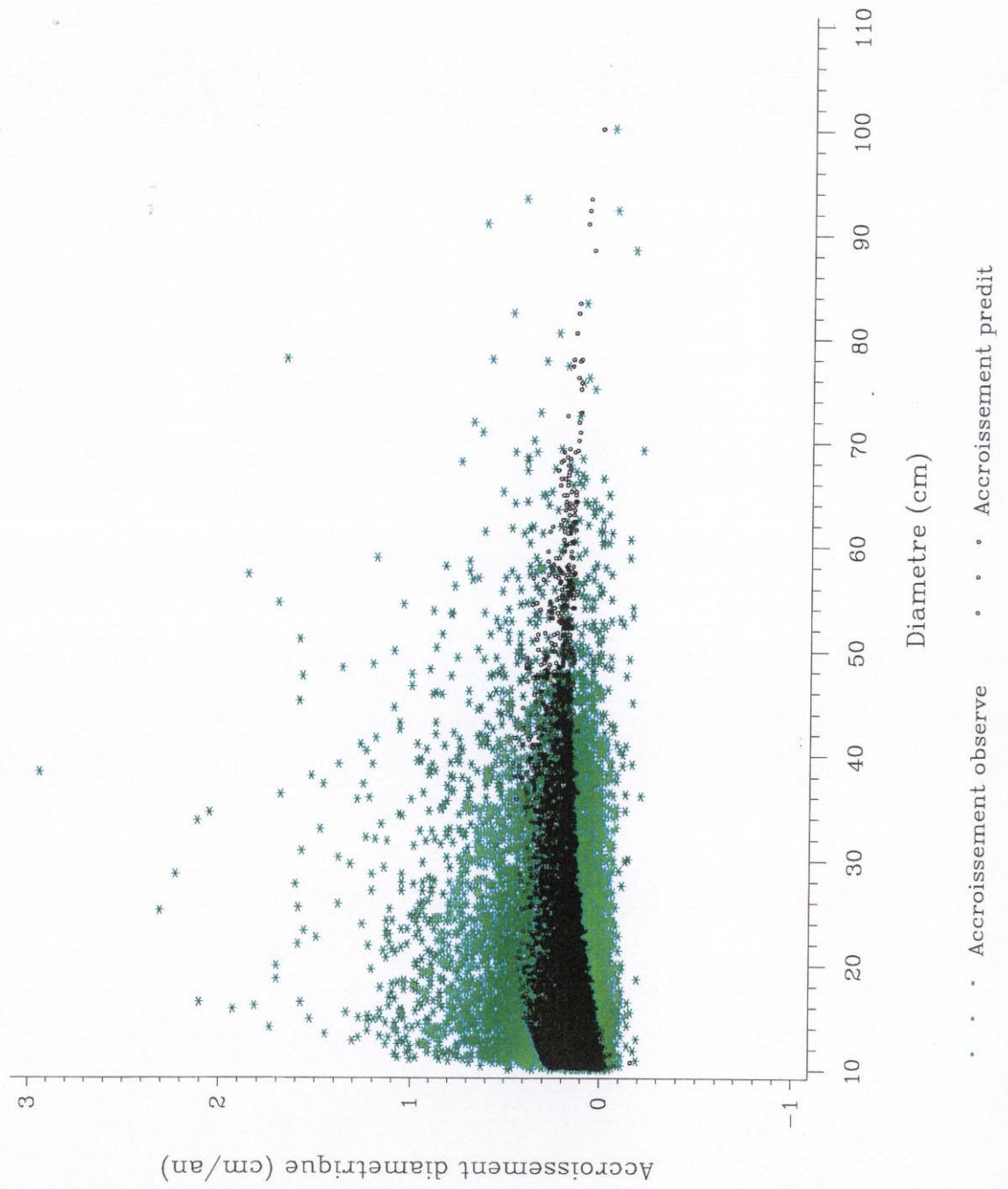


Fig.a12(b). Représentation des accroissements moyens annuels observés sur les 10745 individus du fichier de calibration en fonction de l'indice de compétition *NBD*, avec superposition des valeurs prédites par le modèle [13].

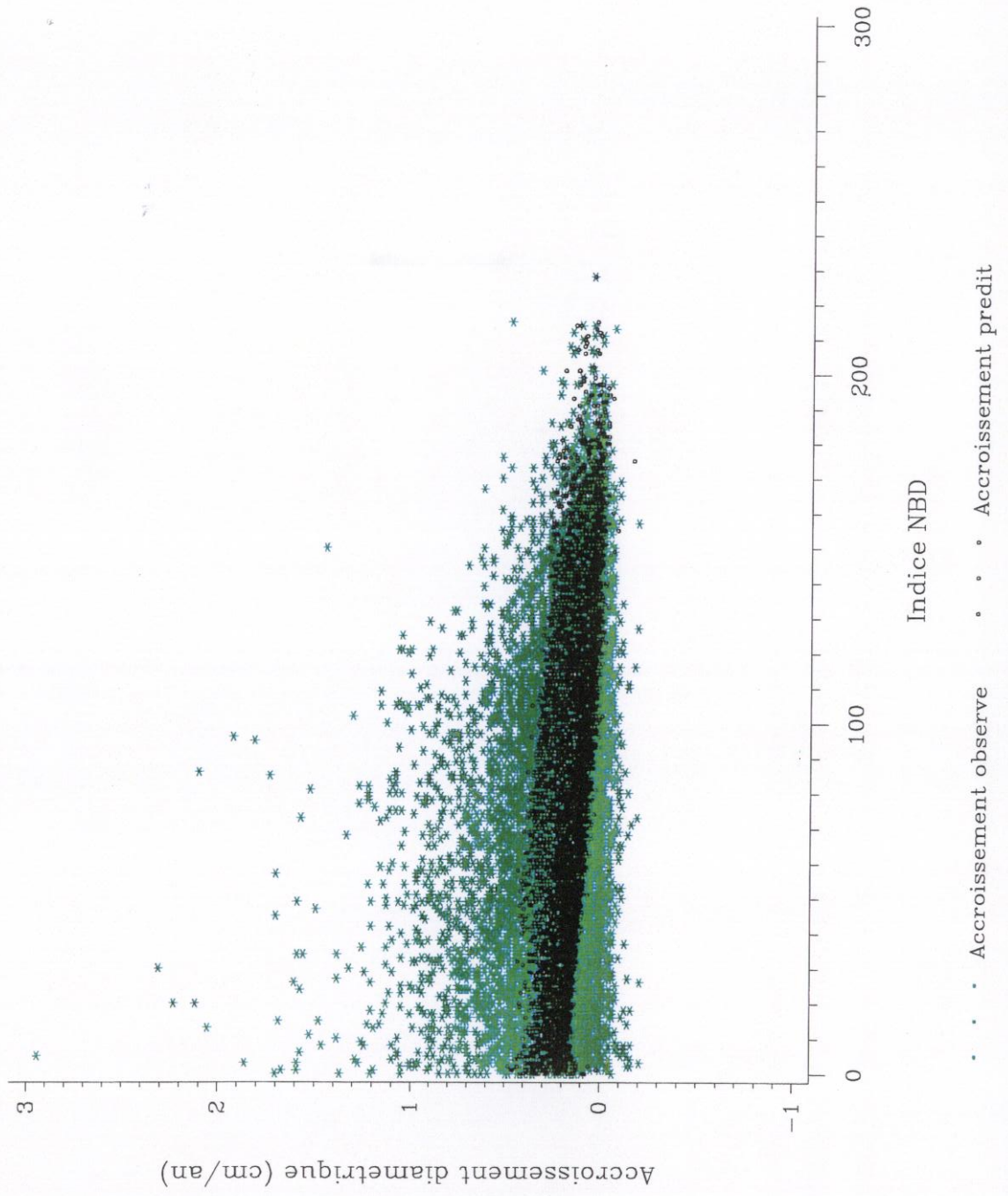
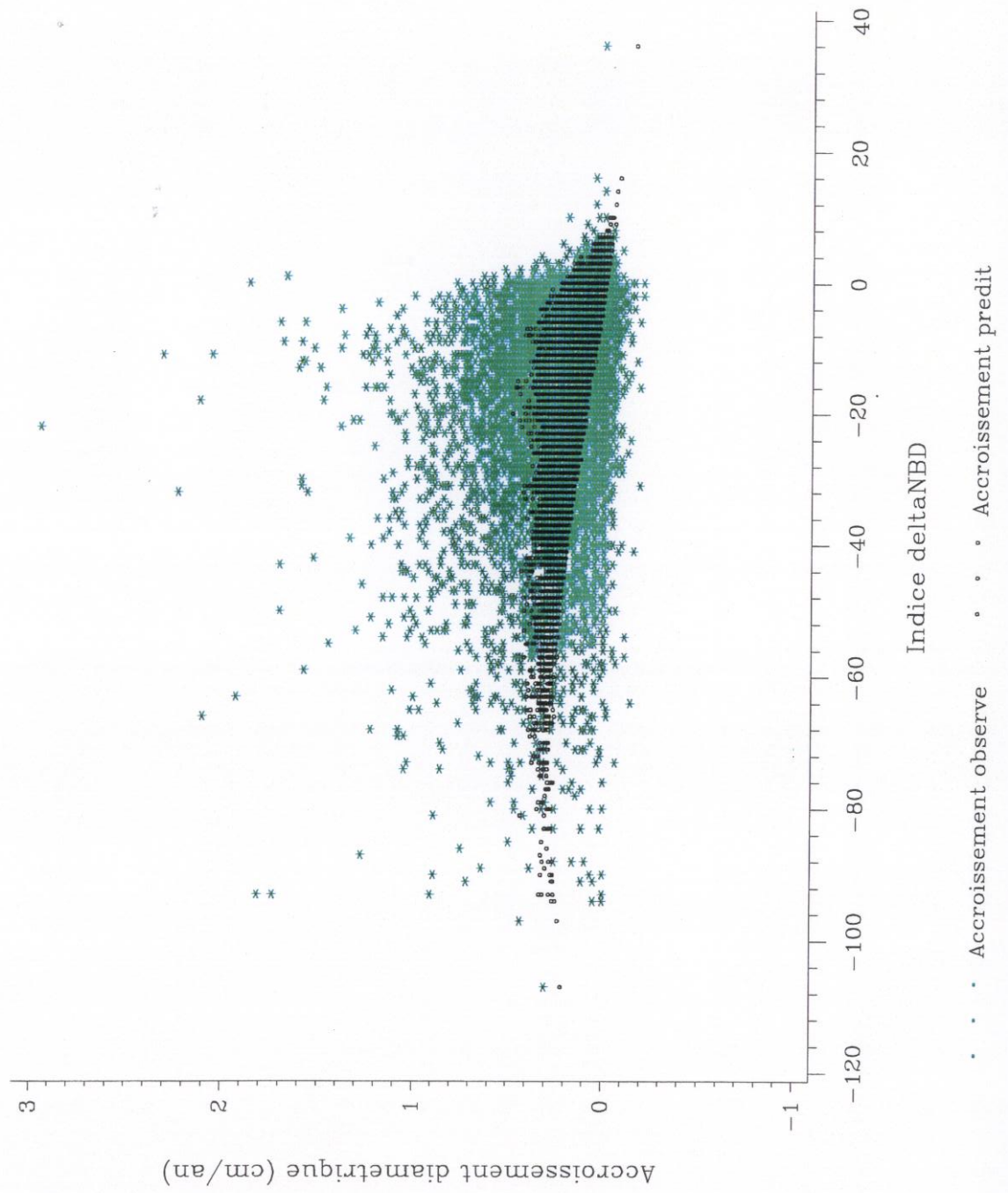


Fig. a12(c). Représentation des accroissements moyens annuels observés sur les 10745 individus du fichier de calibration en fonction de l'indice de compétition ΔNBD , avec superposition des valeurs prédites par le modèle [13].



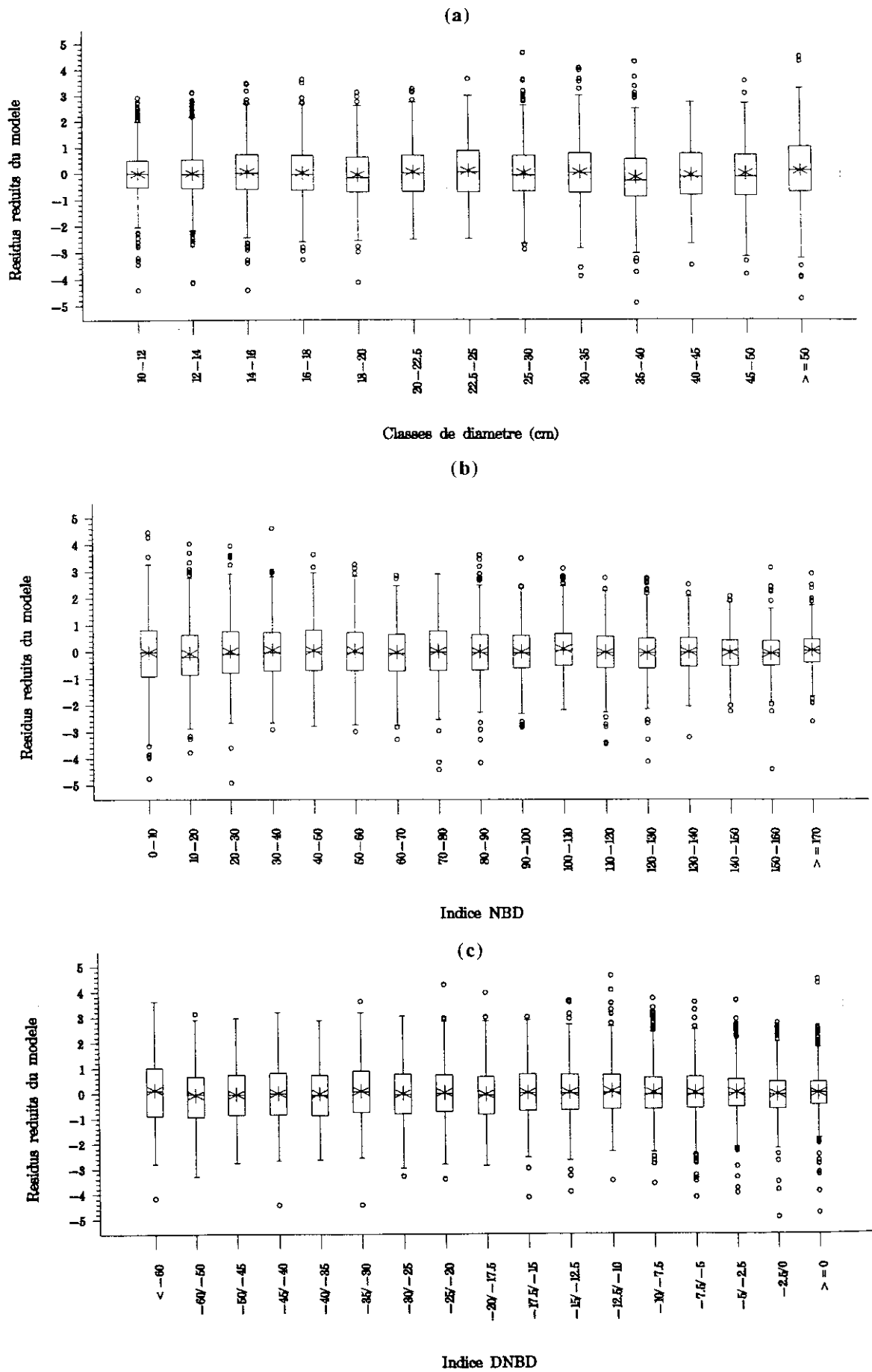


Fig. a 13. Représentation des résidus réduits du modèle [13] en fonction des trois variables explicatives : (a) diamètre à 1m30 ; (b) indice *NBD* ; (c) indice ΔNBD . 10745 arbres sont concernés.

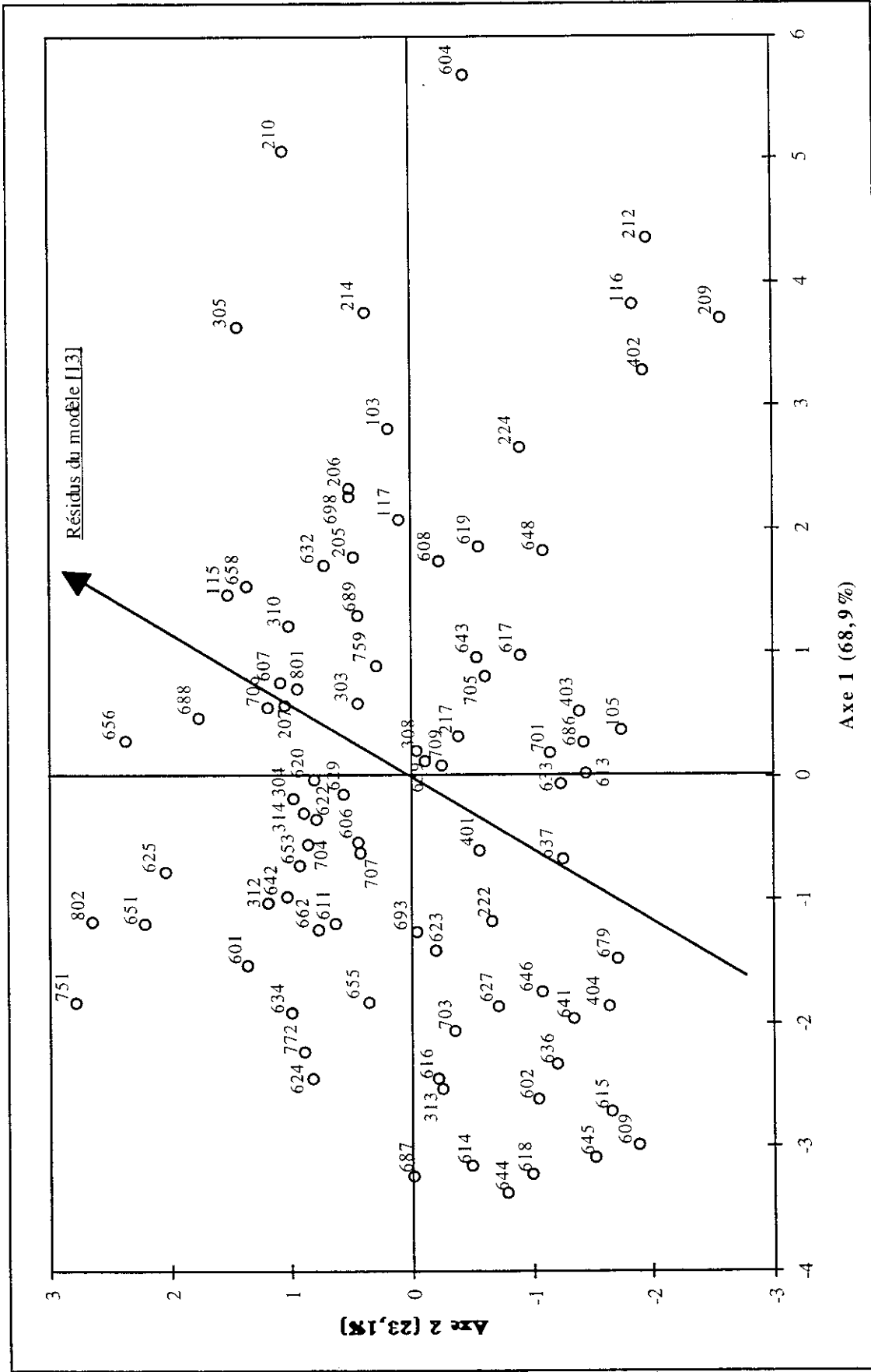


Fig.a14. Projection des 87 espèces étudiées sur le premier plan factoriel de l'ACP (cf. légende de la figure 4.13, texte principal). **NB.** On se reportera à la liste d'espèces fournie au Tableau a8 (même annexe) pour l'identification des codes.

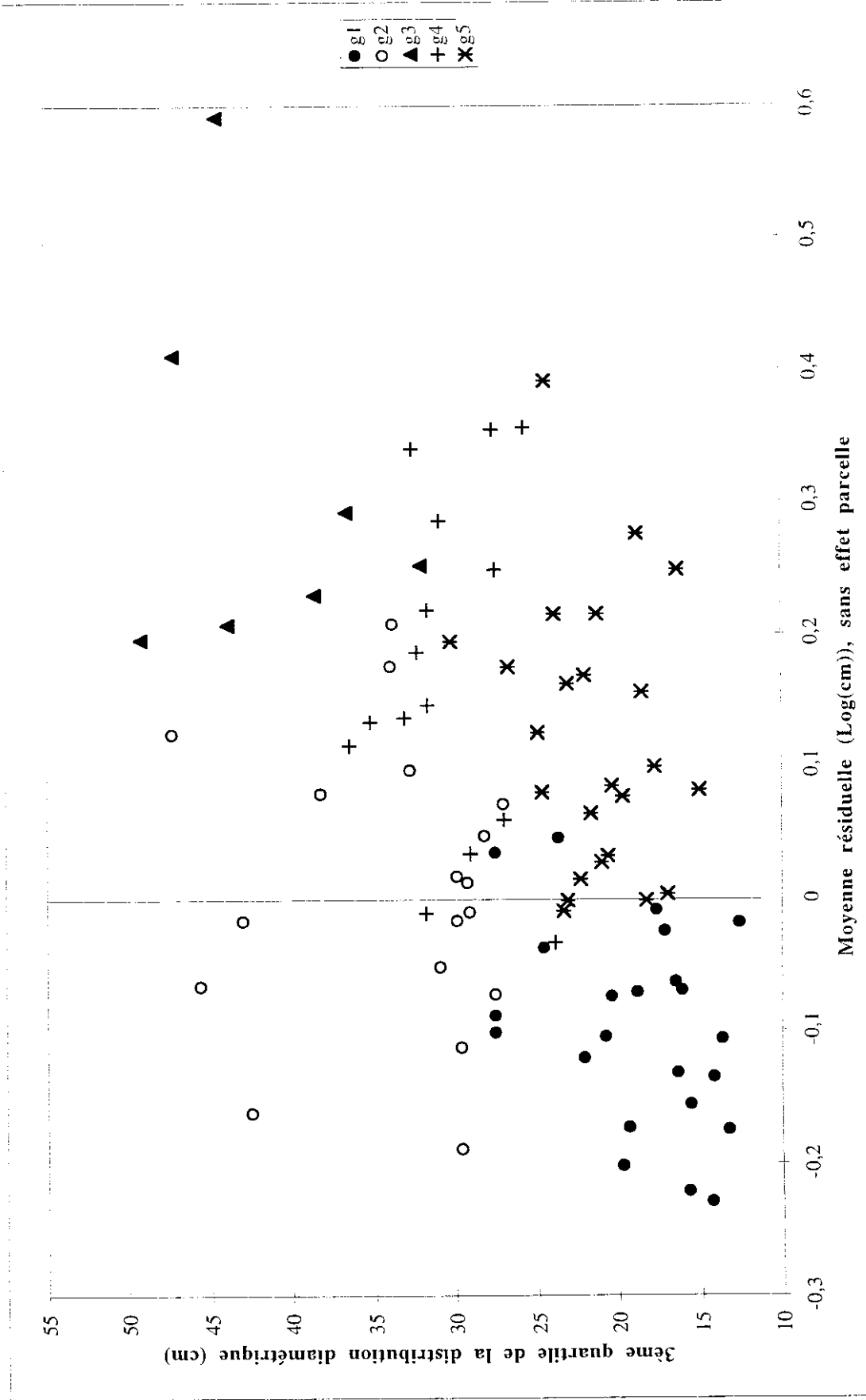


Fig.a15. Distribution des 87 espèces (ou groupes d'espèces) les mieux représentées dans le jeu de données d'ajustement, en fonction : (i) de la moyenne des résidus du modèle de croissance [13], après élimination de l'effet parcelle au moyen d'une ANOVA à un facteur ; et (ii) du 3ème quartile de la distribution diamétrique de l'ensemble des individus vivants en 1988 sur les 12 parcelles. On a représenté l'appartenance de chaque espèce à cinq nouveaux groupes. Espèces dont la sensibilité apparente au statut social est faible à nulle : g1 (espèces de taille diamétrique potentielle faible) et g2 (espèces de taille moyenne à grande). Espèces dont la sensibilité apparente au statut social est moyenne à forte : g3 (espèces de grande taille diamétrique potentielle), g4 (espèces de taille moyenne à grande) et g5 (espèces de petite taille). Voir codes espèces sur la Fig.a16 suivante. 9973 individus pris en compte.

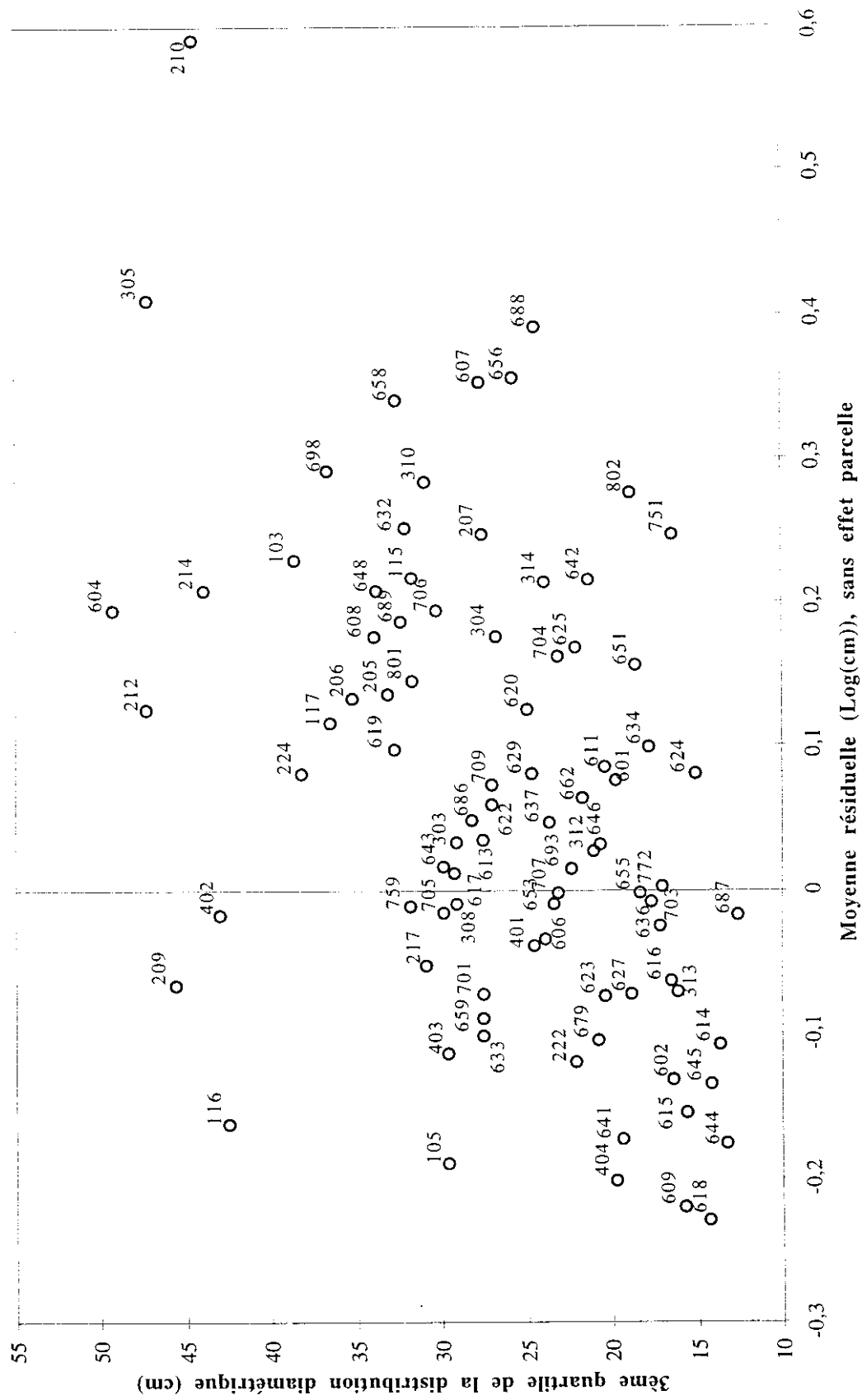


Fig.a16. Distribution des 87 espèces (ou groupes d'espèces) les mieux représentées dans le jeu de données de calibration, en fonction : (i) de la moyenne des résidus du modèle de croissance [13], après élimination de l'effet parcelle au moyen d'une ANOVA à un facteur ; et (ii) du 3ème quartile de la distribution diamétrique de l'ensemble des individus vivants en 1988 sur les 12 parcelles. 9973 individus sont pris en compte. **NB.** On se reportera à la liste d'espèces fournie au Tableau a8 (même annexe) pour l'identification des codes.

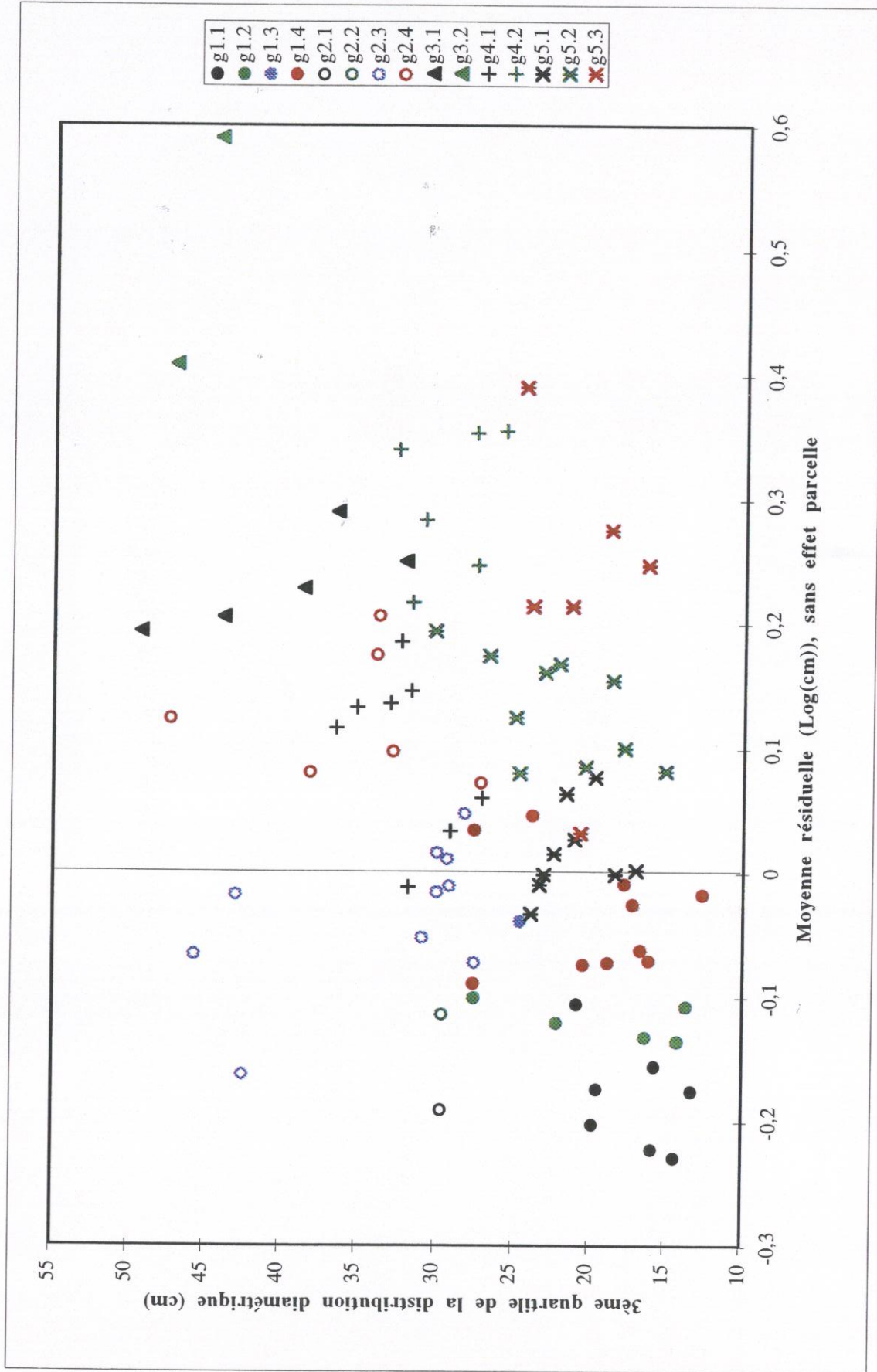


Fig.a17. Distribution des 87 espèces (ou groupes d'espèces) les mieux représentées dans le jeu de données de calibration, en fonction : (i) de la moyenne des résidus du modèle de croissance [13], après élimination de l'effet parcelle au moyen d'une ANOVA à un facteur ; et (ii) du 3ème quartile de la distribution diamétrique de l'ensemble des individus vivants en 1988 sur les 12 parcelles. On a représenté l'appartenance de chaque espèce à nos 15 groupes de croissance. g1, g2, g3, g4, g5 : voir légende de la figure a15. Subdivisions : voir texte principal. 9973 individus sont pris en compte.

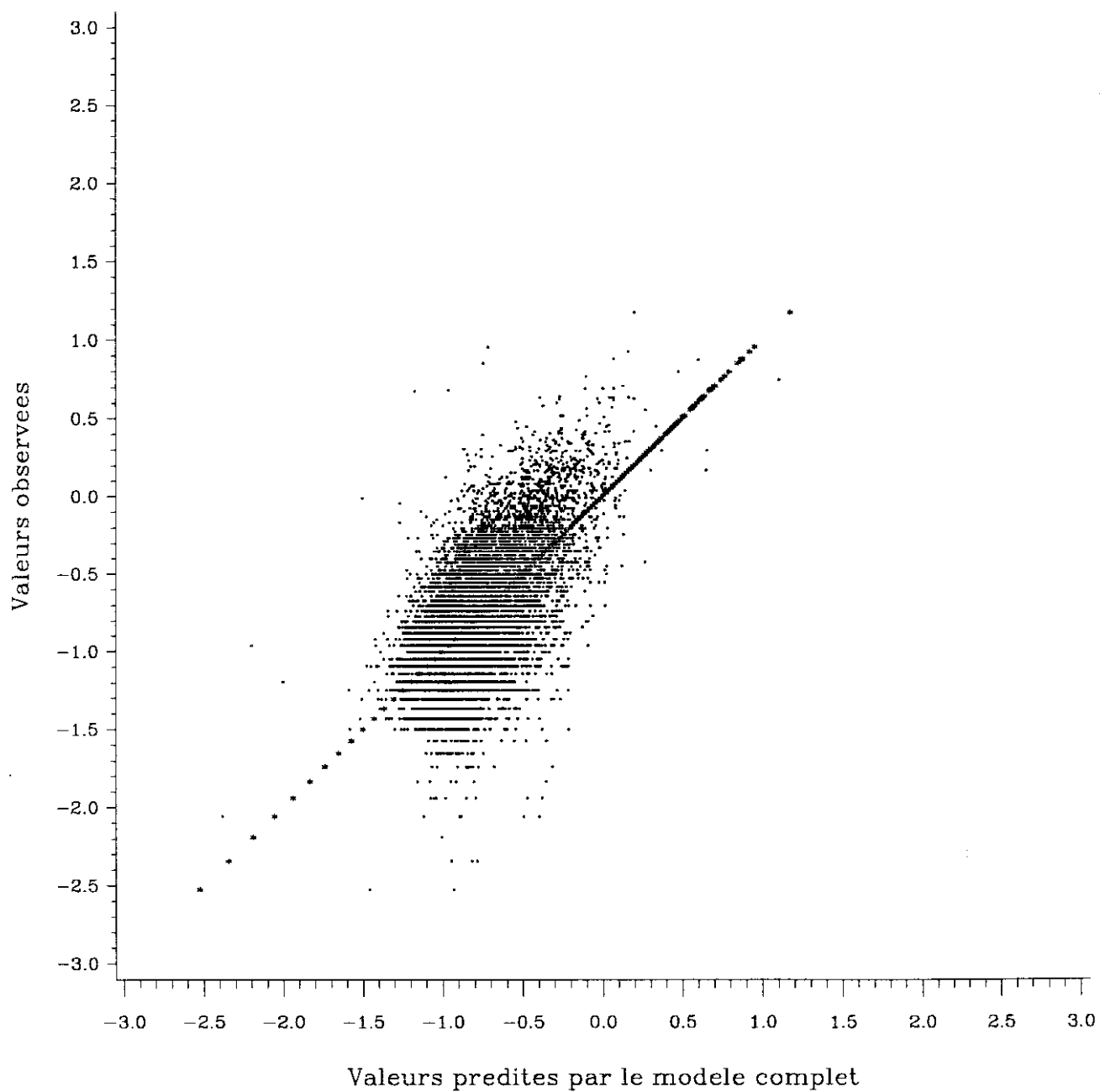


Fig.a18 (a). Illustration de la relation $O_i = aP_i + b_i$ entre les valeurs $O_i = \text{Log}(\Delta D_i + 0,287)$ observées dans le jeu de données de calibration et les valeurs P_i prédites par le modèle complet à 65 paramètres.

Voir Fig.a18 (b) page suivante : même chose avec le jeu de données indépendant.

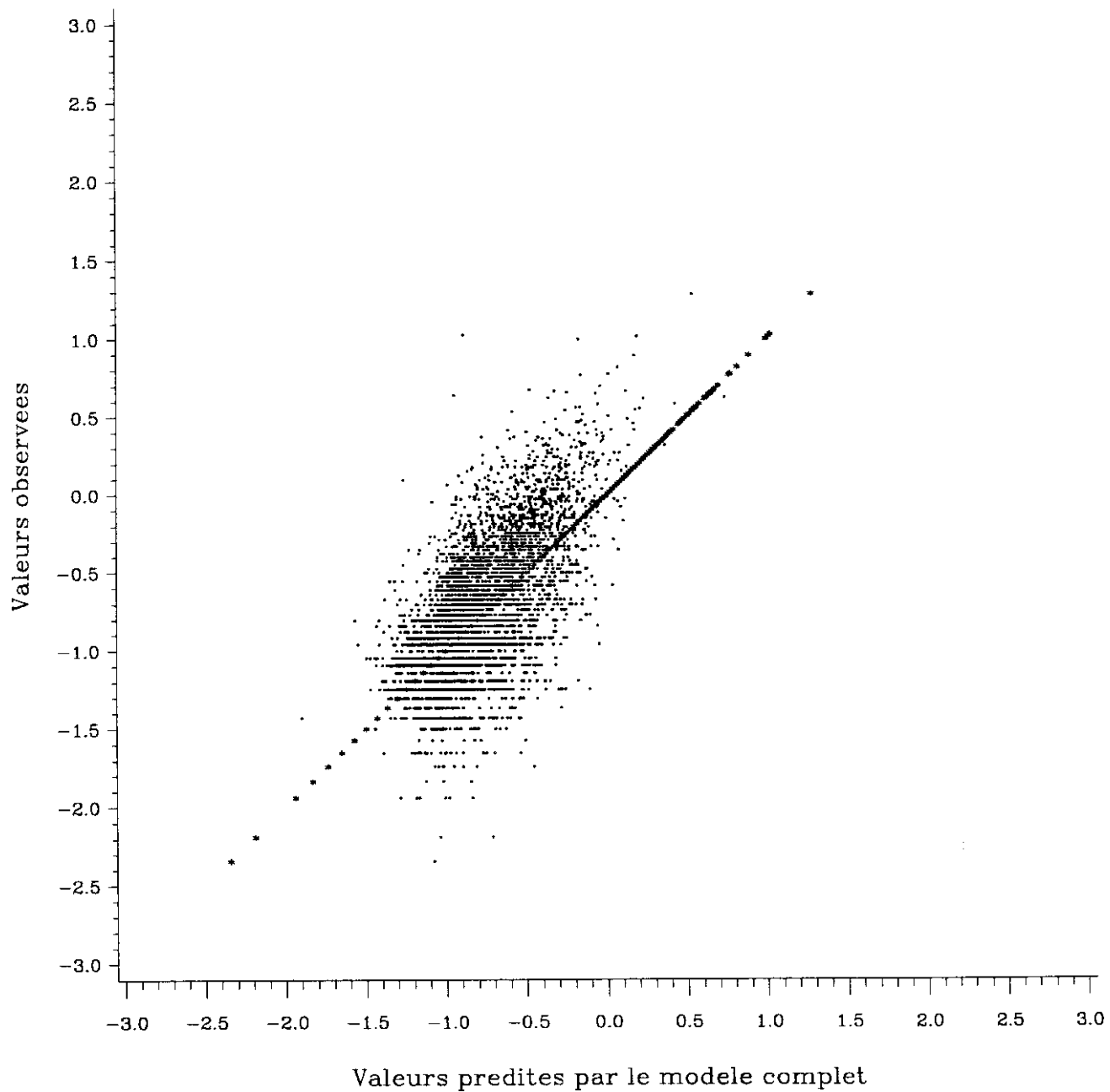


Fig.a18 (b). Illustration de la relation $O_i = aP_i + b_i$ entre les valeurs $O_i = \text{Log}(\Delta D_i + 0,287)$ observées dans le jeu de données indépendant et les valeurs P_i prédites par le modèle complet à 65 paramètres.

Fig.a19 (a). Illustration de la liaison entre les écarts ($O_i - P_i$) calculés respectivement sur la période 1988-1991 et 1991-1994 sur les parcelles de calibration initiales. Le modèle de croissance utilisé est le modèle complet à 65 paramètres prenant en compte les groupes d'espèces.

Voir Fig.a19 (b) page suivante : même chose avec le modèle [13].

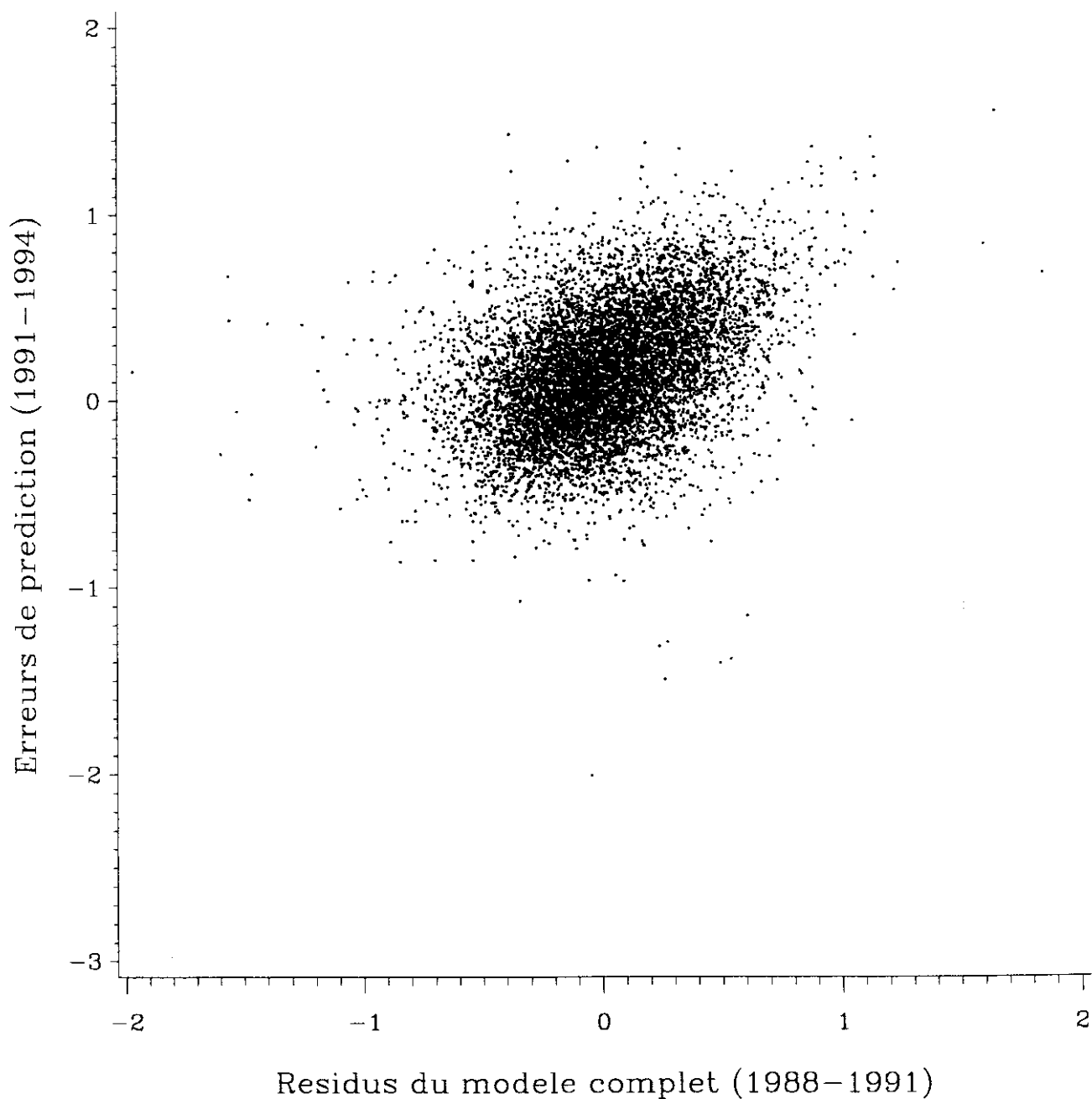
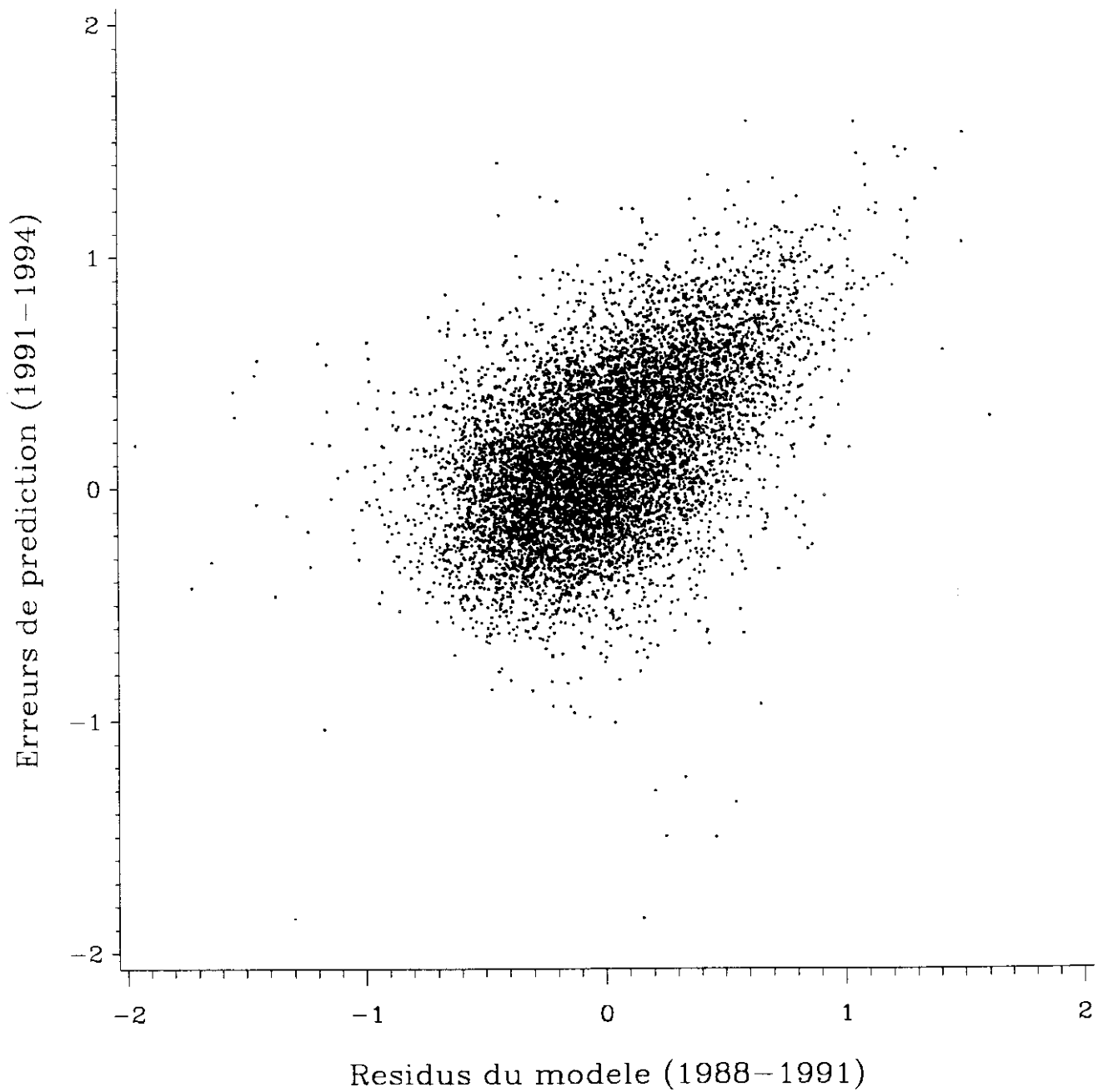


Fig.a19 (b). Illustration de la liaison entre les écarts ($O_i - P_i$) calculés respectivement sur la période 1988-1991 et 1991-1994 sur les parcelles de calibration initiales. Le modèle de croissance utilisé est le modèle [13].



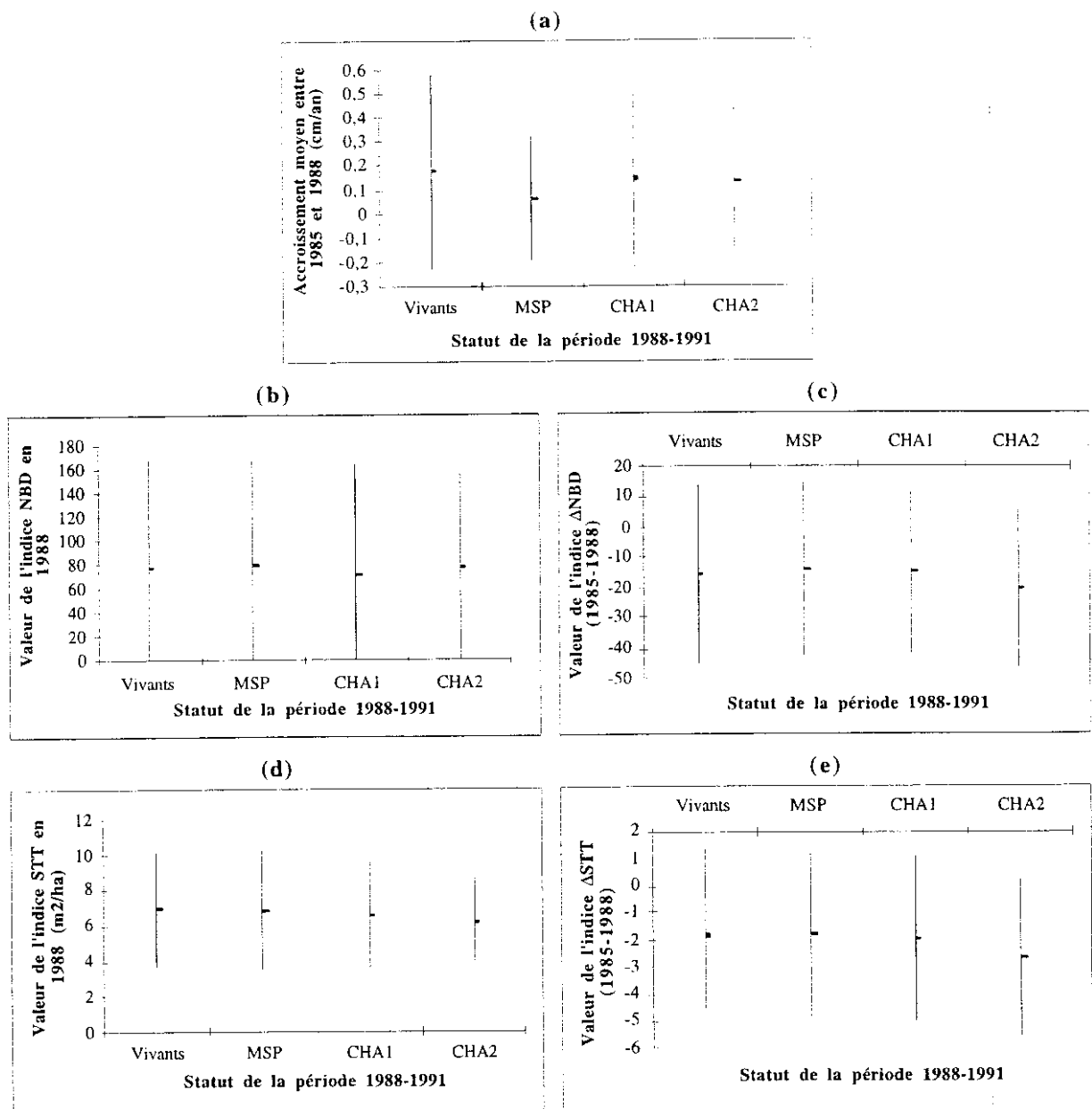


Fig.a20. Etude de l'influence de différentes variables potentiellement explicatives de la mortalité. Les individus sont répartis en quatre catégories en fonction de leur statut individuel observé à l'issue de la période 1988-1991 : vivants, morts sur pied (MSP), chablis primaires (CHA1) ou chablis secondaires (CHA2). Pour chacune de ces catégories, on a représenté la valeur moyenne et l'intervalle de confiance à 95% de : (a) l'accroissement moyen annuel sur le diamètre (cm/an) entre 1985 et 1988 ; (b) la valeur de *NBD* en 1988 ; (c) la valeur de ΔNBD (1985-1988) ; (d) la valeur de *STT* en 1988 et (e) la valeur de ΔSTT (1985-1988).

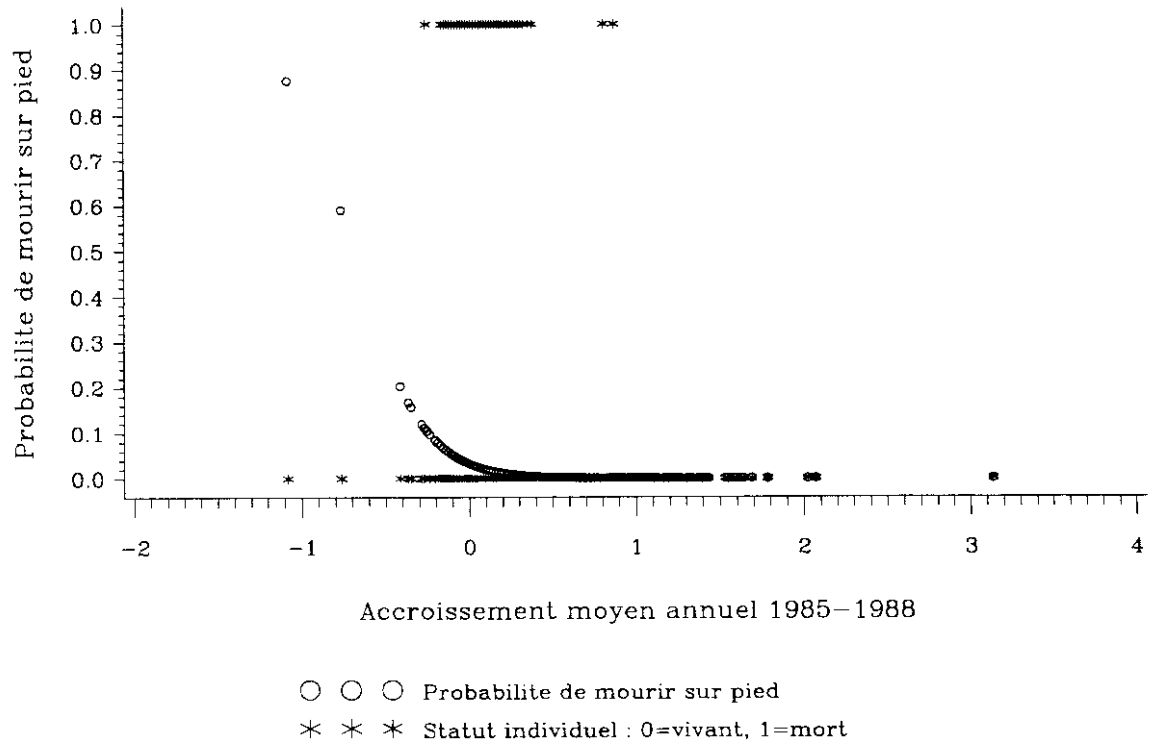


Fig.a21. Comparaison entre le statut (vivant / mort sur pied) des arbres entre 1988 et 1991 et leur probabilité de mourir telle que prédite par le modèle [19].

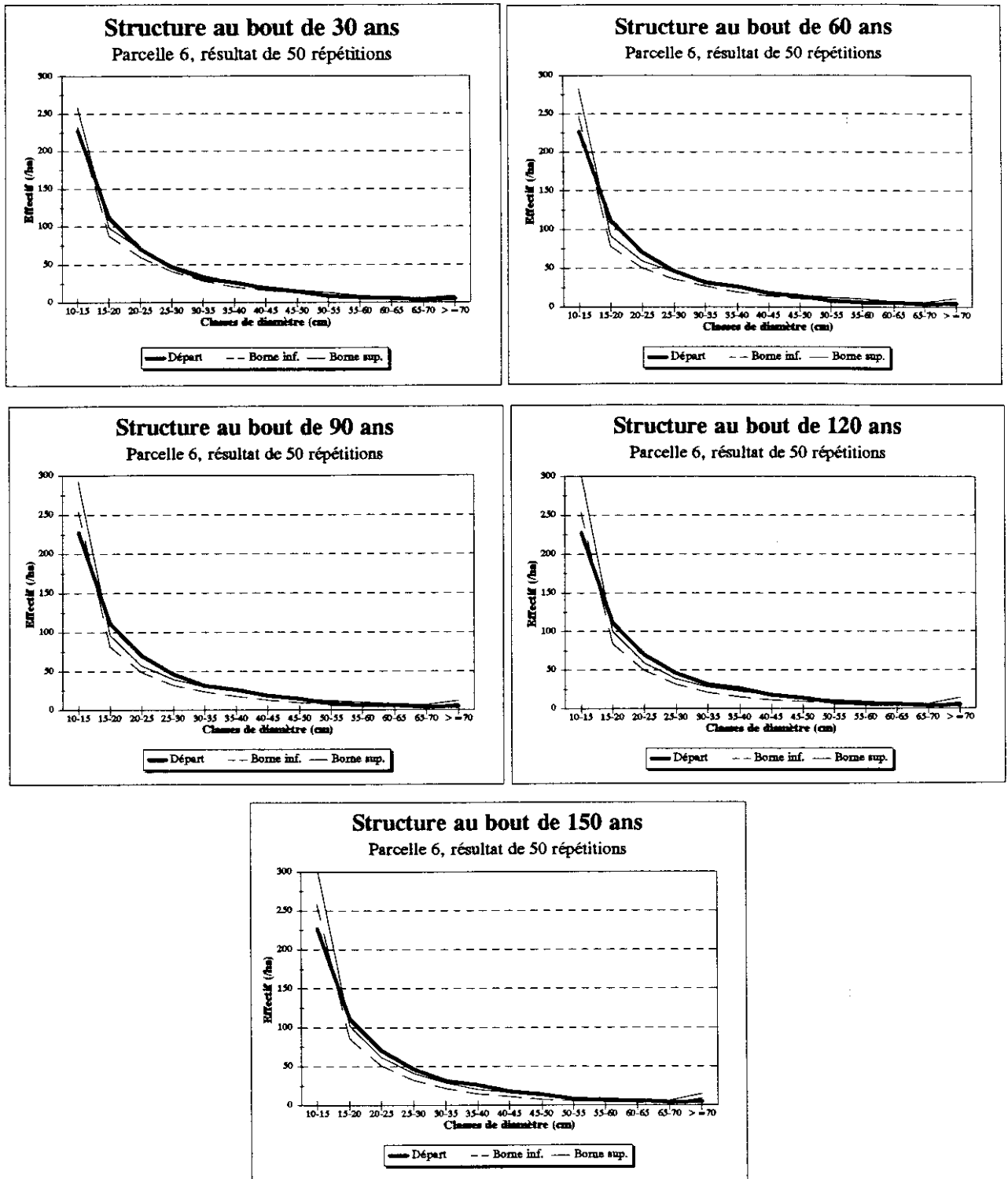


Fig.a22. Etude du comportement du simulateur, à partir de 50 répétitions d'une simulation de 50 cycles (150 ans). La situation initiale est fournie par la parcelle 6, année 1988. Le modèle de croissance est le modèle [14], muni du jeu de paramètres n°1 (voir texte principal). On étudie l'évolution de la structure du peuplement simulé au cours du temps. Sur chaque graphique figurent les effectifs réels observés en 1988 sur la parcelle 6, ainsi que les bornes inférieure et supérieure de l'intervalle de confiance à 95% de la moyenne des effectifs simulés dans chaque classe de diamètre.

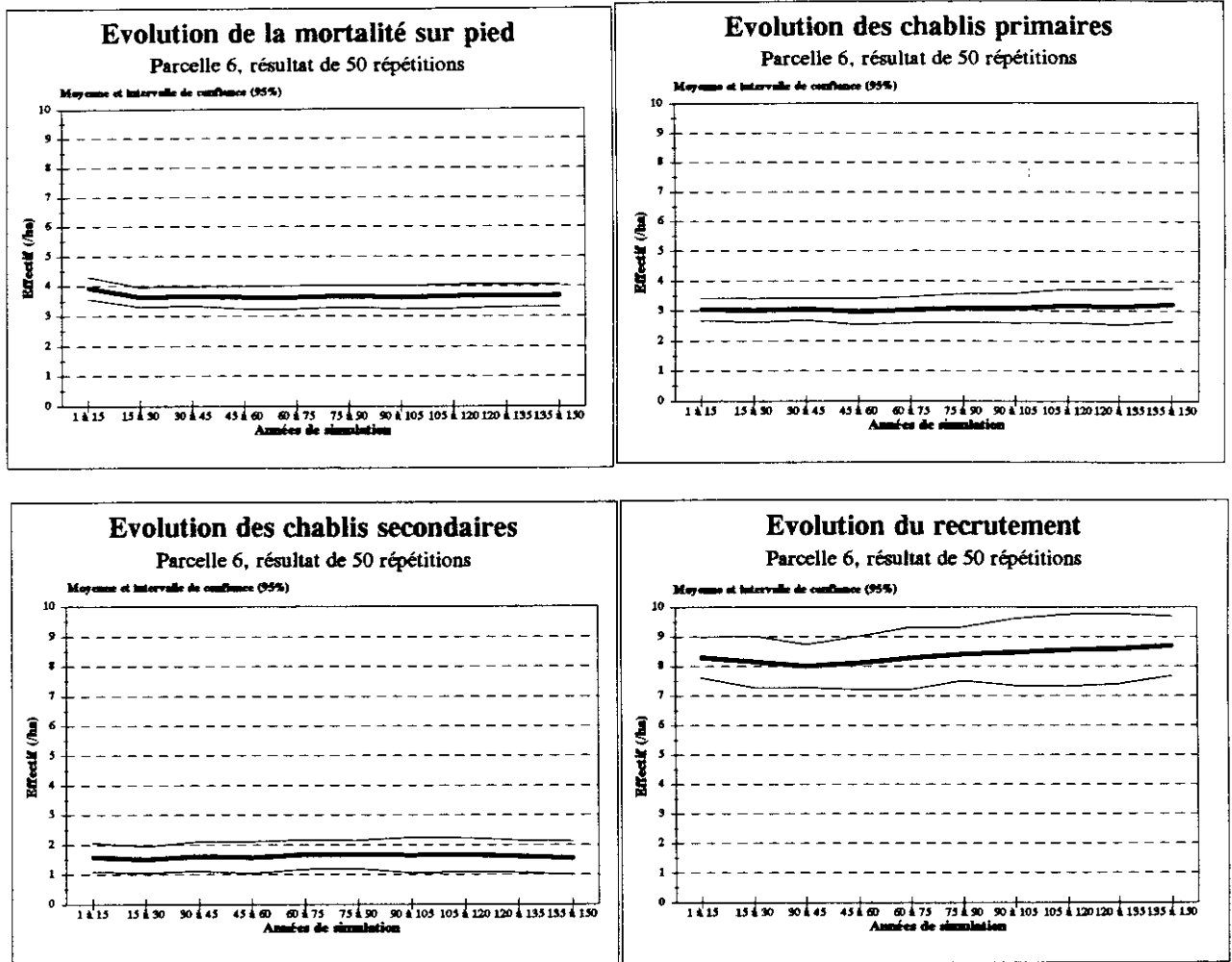


Fig. a23. Etude du comportement du simulateur, à partir de 50 répétitions d'une simulation de 50 cycles (150 ans). La situation initiale est fournie par la parcelle 6, année 1988. Le modèle de croissance est le modèle [14], muni du jeu de paramètres n°1 (voir texte principal). On étudie l'évolution des différentes composantes de la dynamique forestière simulée au cours du temps. Sur chaque graphique figurent la moyenne ainsi que les bornes inférieure et supérieure de l'intervalle de confiance à 95% obtenus par périodes de cinq cycles (15 ans).

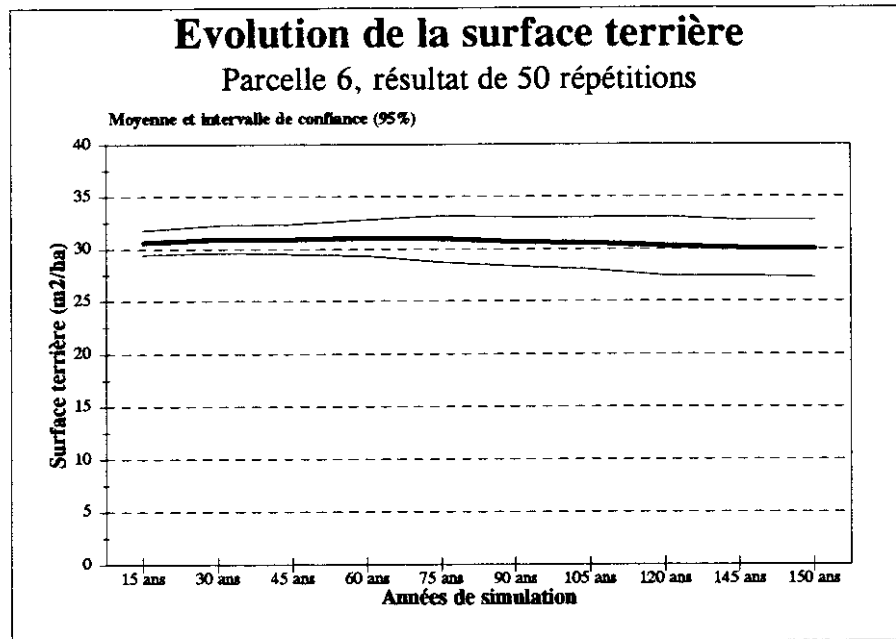
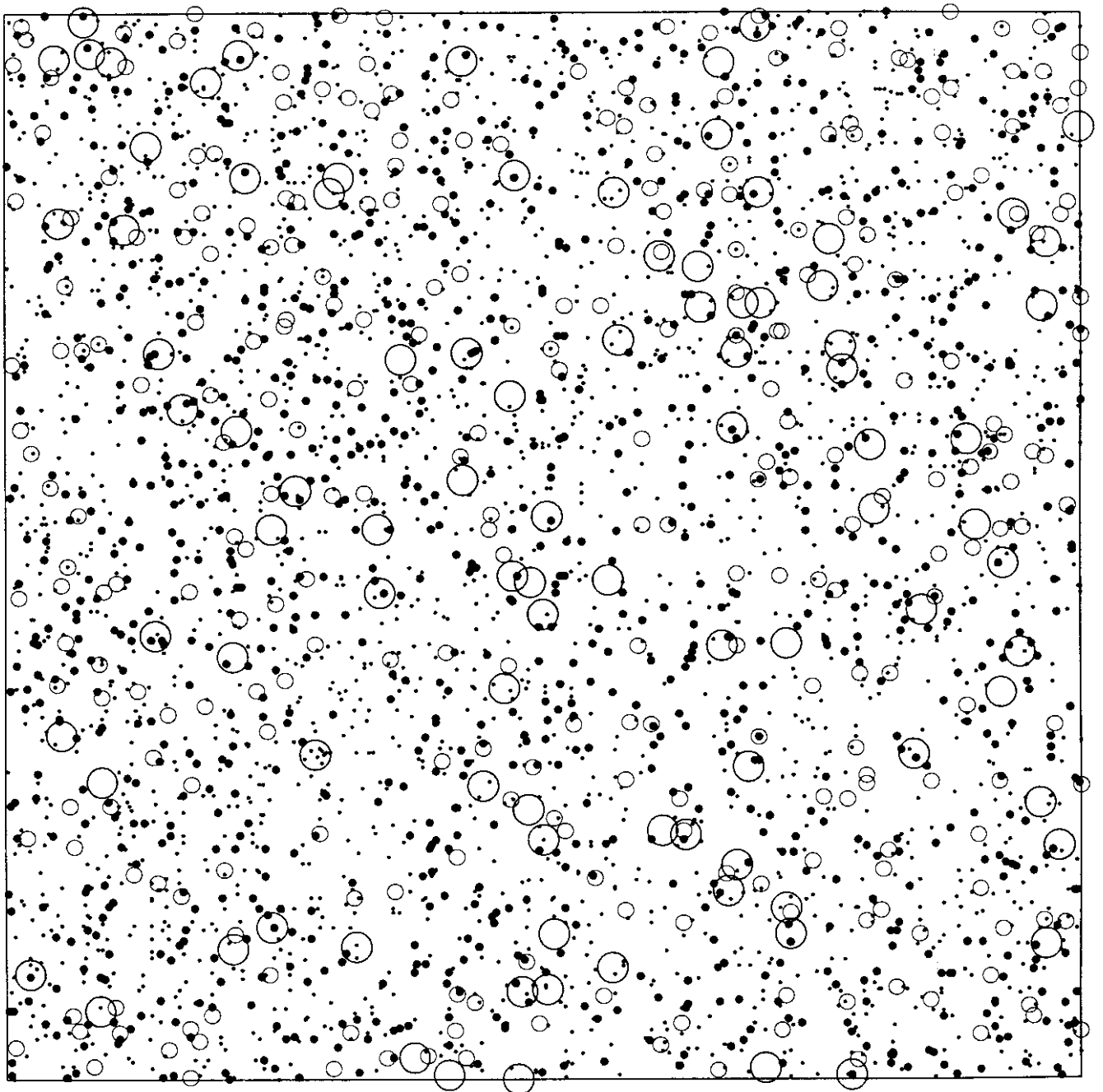


Fig.a24. Etude de la variabilité du simulateur, à partir de 50 répétitions d'une simulation de 50 cycles (150 ans). La situation initiale est fournie par la parcelle 6, année 1988. Le modèle de croissance est le modèle [14], muni du jeu de paramètres n°1 (voir texte principal). On étudie l'évolution de la surface terrière globale au cours du temps. On a fait figurer sur le graphique la moyenne ainsi que les bornes inférieure et supérieure de l'intervalle de confiance à 95% obtenus par périodes de cinq cycles (15 ans).

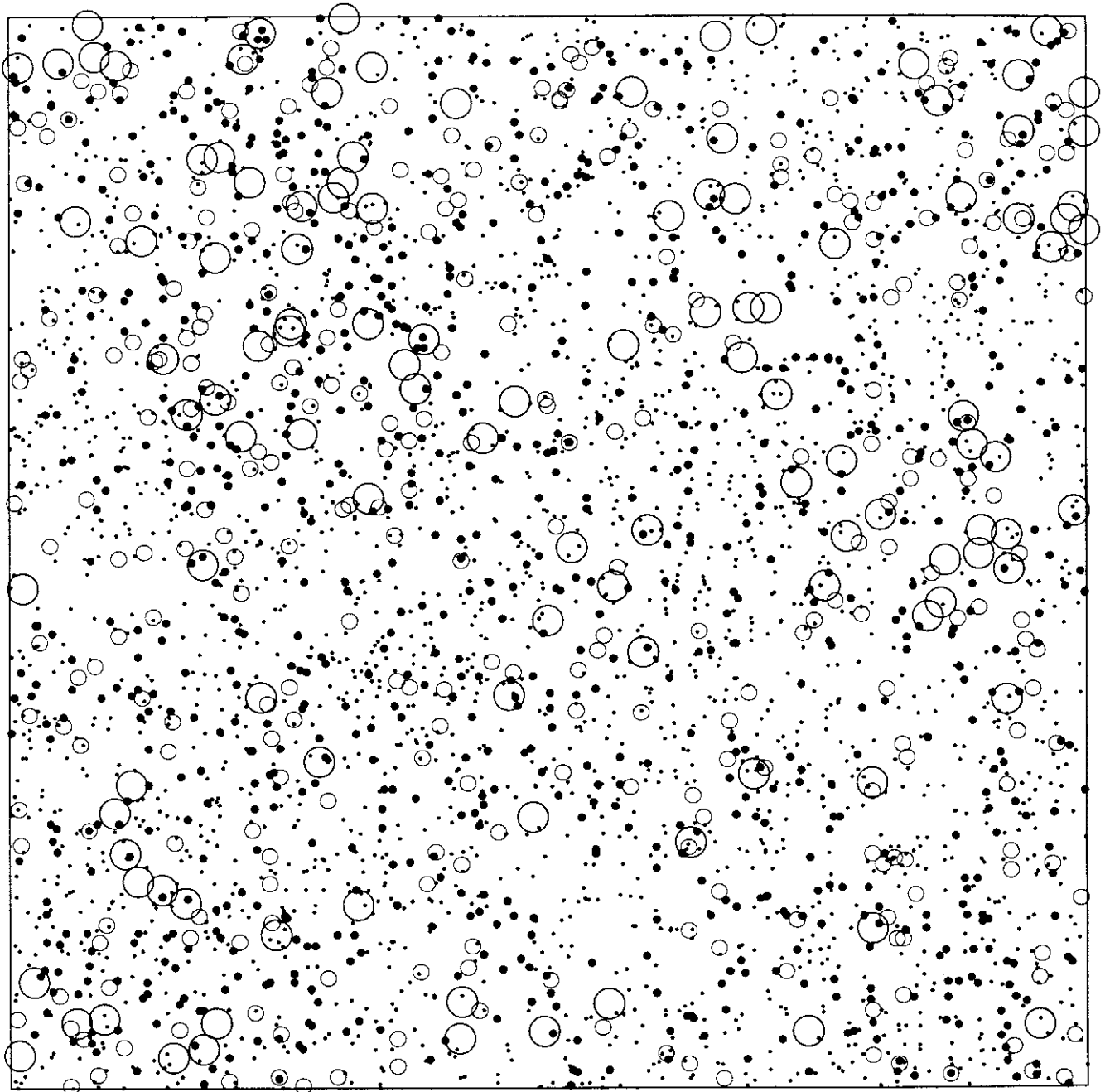


Classes de diamètre (cm)	
·	10 à 20
•	20 à 40
○	40 à 60
○	supérieur à 60

0 ————— 50 m

Parcelle 6 en 1984 (situation réelle).

Fig. a25. Parcelle 6 (témoin) en 1984 : représentation d'une situation réelle.

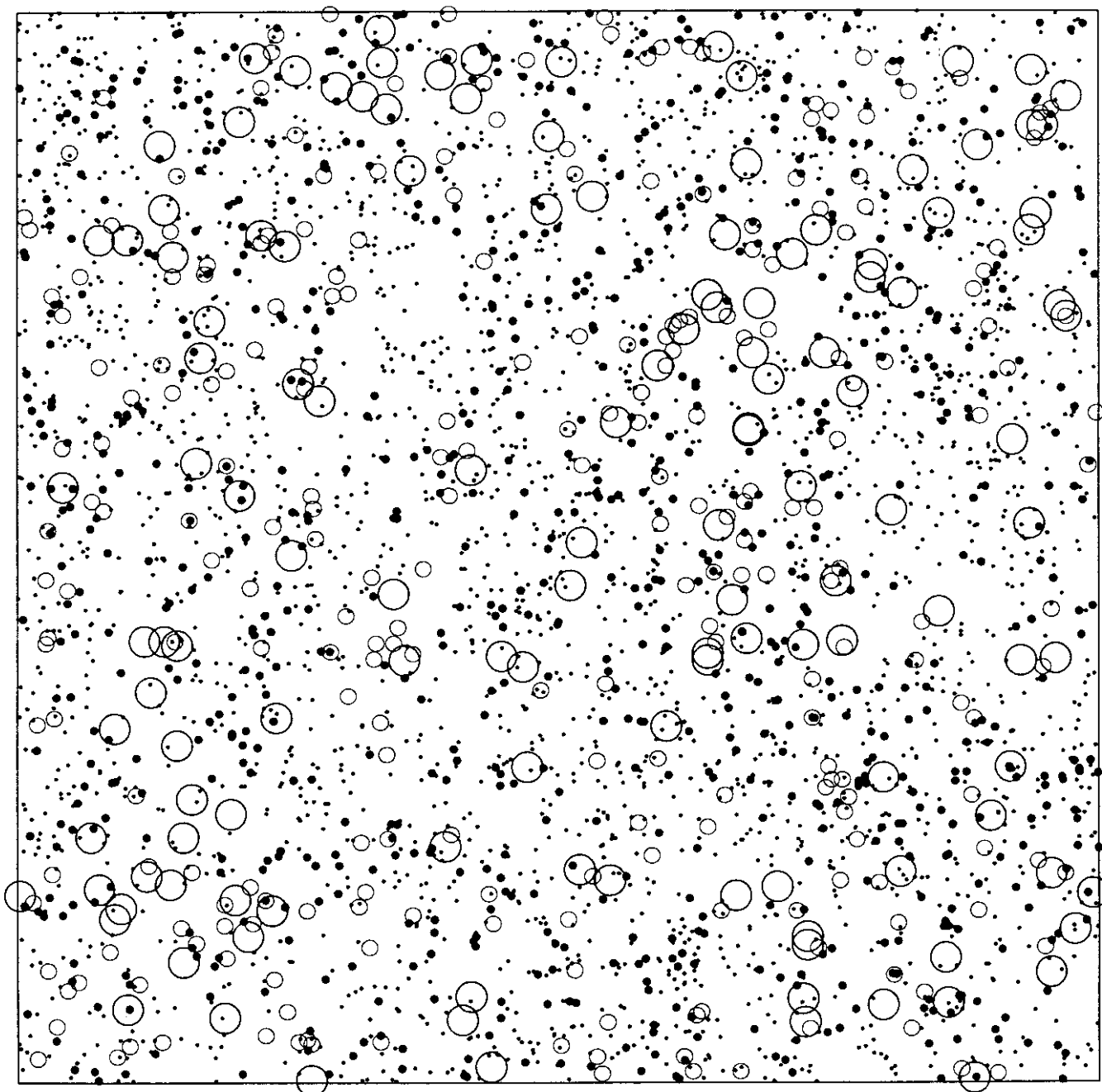


Classes de diamètre (cm)	
·	10 à 20
•	20 à 40
○	40 à 60
○	supérieur à 60

0 50 m

Parcelle 6 au bout de 20 cycles de simulation (60 ans).

Fig.a26. Peuplement au bout de 20 cycles de simulation (60 ans). La situation initiale est fournie par la parcelle 6 en 1988. Sans perturbation.

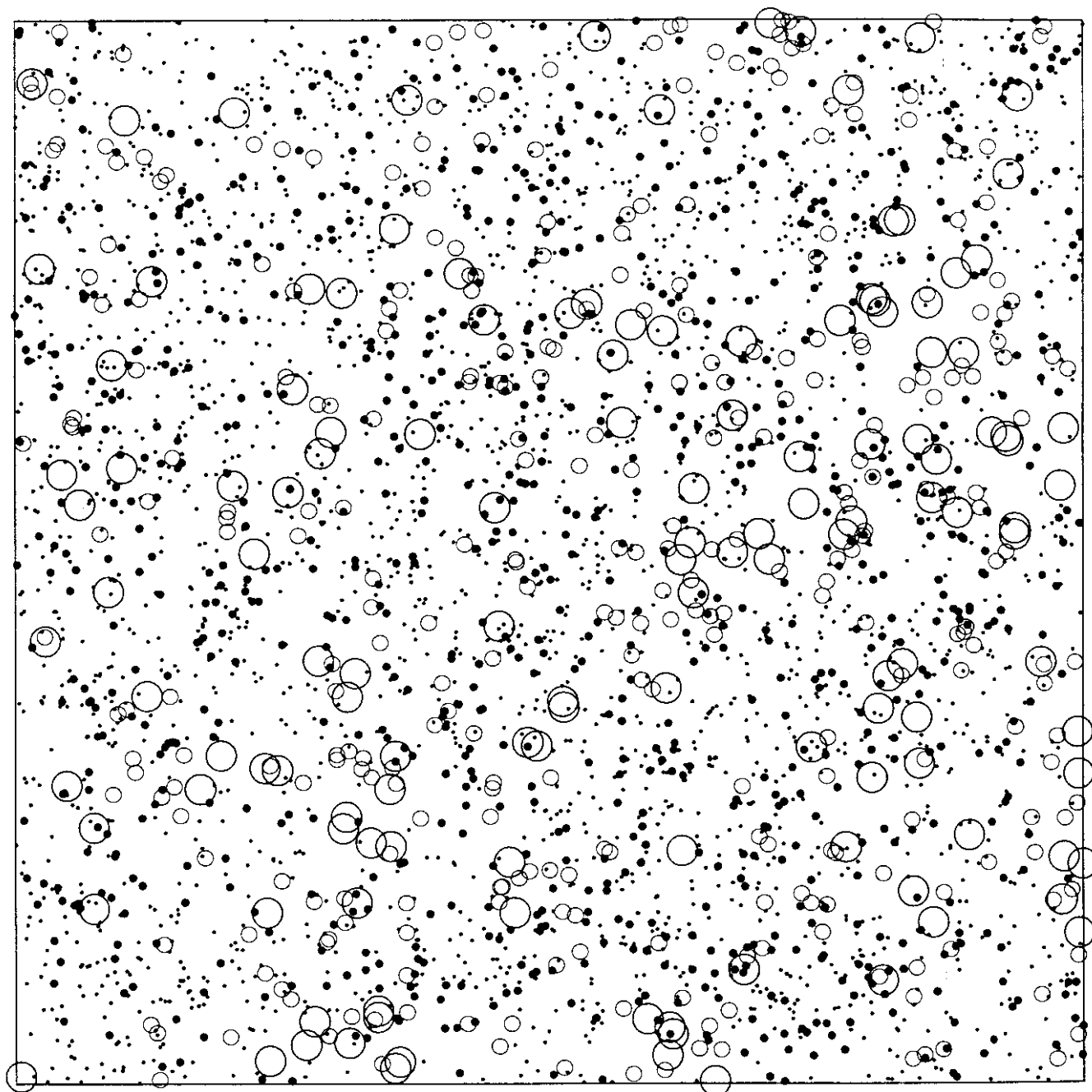


Classes de diamètre	
·	10 à 20
•	20 à 40
○	40 à 60
○	Supérieur à 60

0 ————— 50 m

Parcelle 6 au bout de 50 cycles de simulation (150 ans)

Fig. a27. Peuplement au bout de 50 cycles de simulation (150 ans). La situation initiale est fournie par la parcelle 6 en 1988. Sans perturbation.

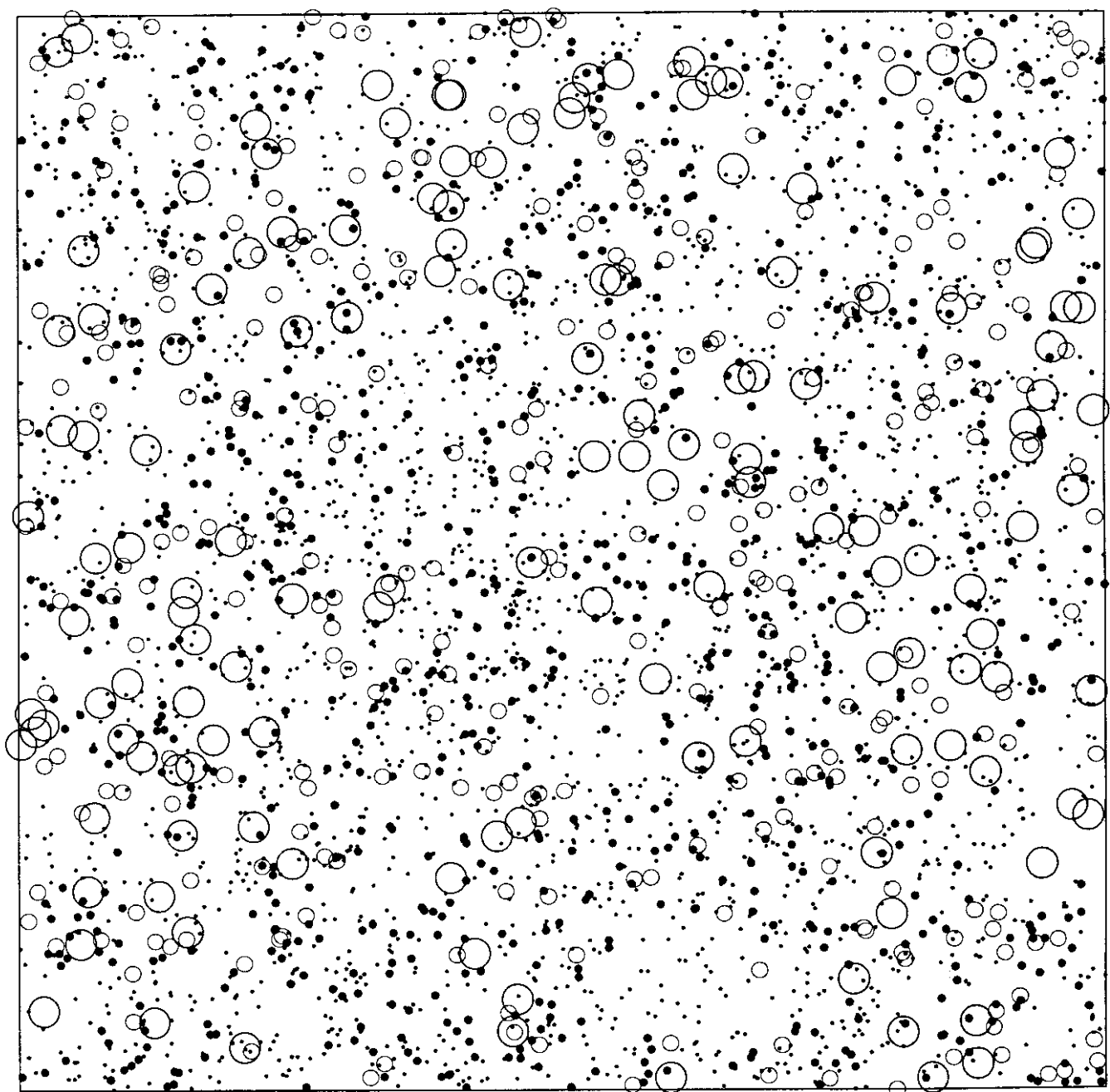


Classes de diamètre	
·	10 à 20
•	20 à 40
○	40 à 60
○	supérieur à 60

0 50 m

Parcelle 6 au bout de 100 cycles de simulation (300 ans)

Fig. a28. Peuplement au bout de 100 cycles de simulation (300 ans). La situation initiale est fournie par la parcelle 6 en 1988. Sans perturbation.



Classes de diamètre	
·	10 à 20
•	20 à 40
○	40 à 60
○	supérieur à 60

0 50 m

Parcelle 6 au bout de 200 cycles de simulation (600 ans)

Fig. a29. Peuplement au bout de 200 cycles de simulation (600 ans). La situation initiale est fournie par la parcelle 6 en 1988. Sans perturbation.

Tableau a.1. Table de contingence utilisée pour les AFC du paragraphe 1.2.3.2. Individus vivants à la campagne 5 (1988), appartenant aux 87 espèces les mieux représentées dans les deux jeux de données étudiés (ajustement et validation). Fin du tableau page suivante.

Code espèce	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	Total
403	126	59	30	82	135	147	122	58	97	169	186	201	1412
404	150	181	81	91	79	89	98	68	123	140	111	60	1271
401	42	24	57	50	88	144	60	31	52	46	159	60	813
224	0	0	3	40	68	133	99	33	18	14	110	59	577
614	46	45	90	44	48	49	46	28	23	43	70	37	569
703	27	15	15	34	80	71	37	34	52	25	98	59	547
705	6	59	25	49	82	6	56	33	121	22	16	56	531
701	99	37	5	43	57	41	26	12	8	12	72	13	425
704	54	22	37	34	31	14	22	19	23	35	76	40	407
105	39	21	30	23	36	21	26	31	37	27	58	36	385
222	60	38	36	15	20	37	38	25	17	38	15	21	360
607	43	46	20	21	47	33	37	9	19	15	28	13	331
616	26	30	32	25	23	39	33	8	16	11	35	18	296
629	31	25	10	19	21	20	20	16	32	22	41	25	282
698	6	0	0	0	40	48	7	15	41	6	80	19	262
604	45	28	23	22	1	0	33	14	2	0	66	8	242
402	5	4	8	20	24	17	30	22	18	17	45	24	234
207	35	17	16	9	15	18	29	9	12	19	26	22	227
633	14	30	25	8	18	11	10	13	19	15	41	23	227
709	48	33	17	20	28	4	29	18	16	1	6	3	223
602	19	24	20	6	12	26	31	9	16	25	20	13	221
314	42	14	16	8	26	22	11	10	6	34	22	9	220
103	13	6	8	17	4	16	6	20	45	14	20	23	192
662	4	24	10	15	7	12	26	5	29	11	24	6	173
618	28	11	9	9	12	13	15	12	13	26	17	6	171
308	15	6	10	6	17	14	10	4	28	13	33	6	162
606	25	13	20	6	8	13	10	10	7	6	26	9	153
707	14	15	27	10	7	10	19	7	6	12	11	6	144
655	16	12	8	8	6	11	11	7	17	9	16	18	139
617	5	11	5	6	10	12	15	7	14	3	31	7	126
310	8	9	3	7	12	16	10	6	18	14	14	8	125
641	6	10	8	7	13	7	10	5	11	8	18	10	113
643	7	8	7	7	10	7	10	8	16	11	4	16	111
627	10	3	1	6	1	6	8	7	16	9	26	16	109
117	10	2	0	8	24	3	11	7	13	6	10	9	103
206	4	2	1	10	9	3	18	12	17	11	8	6	101
608	0	0	0	7	9	16	6	0	11	7	29	11	96
679	5	8	6	12	22	11	8	2	10	0	7	1	92
619	9	5	7	9	8	6	3	7	5	3	19	6	87
658	4	6	3	2	10	1	6	14	5	10	15	9	85
313	4	7	3	4	3	6	6	6	12	9	13	11	84
687	1	0	0	0	3	74	3	0	0	0	0	3	84
653	11	3	6	7	9	3	7	5	6	1	20	4	82
623	8	5	8	3	8	9	8	2	8	7	9	6	81
644	2	1	3	0	7	6	9	2	9	10	14	17	80
312	27	7	4	1	3	10	8	2	7	2	1	5	77
634	7	12	12	2	2	0	4	4	5	11	11	3	73
214	14	9	6	6	2	2	8	3	7	6	6	2	71

636	13	21	5	2	10	3	6	3	0	2	2	3	70
620	1	10	4	7	6	0	6	4	5	11	6	4	64
759	7	6	2	4	0	13	9	6	7	0	10	0	64
212	6	7	2	0	6	5	7	4	13	4	6	2	62
304	6	3	7	1	3	9	7	3	2	7	6	5	59
659	6	2	1	14	10	18	4	2	0	1	0	1	59
611	8	12	3	1	6	7	5	3	3	4	4	2	58
609	4	11	2	3	5	1	7	3	2	4	5	10	57
642	7	22	6	1	0	2	1	1	8	3	3	3	57
706	8	5	1	4	5	2	3	3	8	3	9	5	56
622	35	12	4	0	0	1	0	1	1	0	1	0	55
601	6	4	0	3	3	10	5	1	8	4	5	3	52
210	3	8	2	3	2	9	4	0	4	6	3	6	50
116	7	5	8	0	6	8	0	0	11	4	0	0	49
801	4	5	6	2	4	4	4	4	5	0	10	1	49
802	5	2	1	4	4	12	7	0	4	4	5	0	48
689	4	4	1	2	3	4	10	2	6	3	6	1	46
615	11	7	10	0	4	3	0	1	2	0	6	1	45
613	4	10	2	2	6	2	4	3	1	1	3	5	43
645	2	7	6	6	3	2	2	2	1	4	1	6	42
205	0	3	1	2	3	1	1	0	14	3	9	4	41
115	2	7	2	2	2	3	2	4	6	5	1	4	40
303	3	5	2	0	5	3	2	6	1	5	3	4	39
646	2	3	12	3	2	3	3	3	1	5	2	0	39
209	6	3	3	2	2	6	1	0	0	3	8	4	38
217	5	0	3	3	3	3	4	1	5	3	6	2	38
625	6	8	2	3	1	1	1	3	1	4	5	2	37
305	4	10	0	1	2	3	2	1	5	1	6	0	35
637	11	2	0	0	4	5	7	0	1	0	4	1	35
693	0	3	5	1	3	0	6	3	5	6	0	3	35
686	2	1	4	2	3	2	3	3	4	1	3	6	34
624	2	0	1	2	4	5	5	0	4	3	1	6	33
632	4	3	2	2	0	6	3	2	4	4	2	1	33
651	3	6	0	1	2	9	2	0	6	1	0	3	33
656	5	5	4	1	0	6	3	0	2	1	2	4	33
648	10	0	2	3	1	0	1	1	2	2	3	7	32
772	4	0	3	0	5	1	4	1	6	4	2	2	32
688	2	3	1	2	4	3	2	1	2	3	3	5	31
751	2	3	1	0	5	8	4	0	3	1	0	4	31
Total	1440	1175	892	927	1342	1460	1302	754	1266	1075	1934	1163	14730

Tableau a2 (4 pages). Classification de quelques modèles de dynamique forestière en fonction de différents critères (degré de fonctionnalité à l'échelle de l'individu, degré d'agrégation des individus, explicitation de la disponibilité des ressources, variables utilisées pour décrire les arbres, objectifs).

MODELES	Degré ^a de fonctionnalité (à l'échelle de l'individu)	Niveau de différenciation de la pression subie par un individu		Global	Explicitation de la disponibilité en ressources			
		« i-state distribution model »	« i-state configuration model »		Implicite ou indices de compétition empiriques	Explicite		
						Lumière	Eau	Nutriments
Peuplement Type Turnbull (1963) ou Lemoine (1995)	1	*	-	-	-	-	-	
Peuplement avec distribution <u>Fonctions de distribution</u> Type Watts (1988), Kuru (1992) Type Kohyama (1991, 1992 a et b) <u>Tables structurées</u> Type FORMIX ^b (Bossel et Krieger, 1994) Type Favrichon (1995) <u>Cohortes</u> Type CAFOGROM (Alder, 1994, 1995)	1 1 4 1 1	* * * * *	- - - (*) (limite) - -	- - - * - - -	- - - - - -	- - - - - -	- - - - - -	
Individuels indépendants de la distance <u>Empiriques</u> Type Stand Prognosis (Wykoff et al. , 1982 et STEMS (Belcher et al. , 1982) Type DIPSIM (Ong et Kleine, 1996) (à suivre)	1 1	* *	(*) (limite, quand les placettes sont petites) - -	- *	- -	- -	- -	

Tableau a2 (suite).

MODELES	Degré ^a de fonctionnalité (à l'échelle de l'individu)	Niveau de différenciation de la pression subie par un individu		Explication de la disponibilité en ressources					
		« i-state distribution model »	« i-state configuration model »	Implicite ou indices de compétition empiriques	Explicite				
					Global	Lumière	Eau	Nutriments	
(suite)									
<u>Fonctionnels</u> "Gap models"	3	*	(*) (limite, quand les trouées sont petites)	-	*		*		*
Individuels dépendants de la distance									
<u>Empiriques</u>									
Type Mitchell (1969, 1975)	2	-	*	*		-	-	-	-
Type FOREST (Ek et Monserud, 1974a)	1	-	*	*		-	-	-	-
Type Hubbell et al. (1990)	1	-	*	*		-	-	-	-
Type SILVA (Pretzsch, 1992 a et b)	2	-	*	*		-	-	-	-
Type Courbaud (1991, 1995)	2	-	*	*		-	-	-	-
SELVA (notre modèle)	1	-	*	*		-	-	-	-
<u>Fonctionnels</u>									
Type SORTIE (Pacala et al. (1993)	3	-	*	-		-	*	A terme	A terme
Type SPGROW (Mou et al. , 1993)	4	-	*	-		*	*	*	*
Type ARCADIA (Williams, 1996)	4	-	*	-		-	*	*	*
Modèles architecturaux									
Type AMAPpara (Blaise et de Reffye, 1994)	3/4	-	*	-		-	*	*	*

^a 1 : modèles empiriques. 2 : modèles "semi-empiriques". 3 : modèles "fonctionnels" au sens large. 4 : modèles à base physiologique. Voir texte pour les détails.

^b D'après le classement de Vanclay, 1994, mais FORMIX2 possède de nombreuses caractéristiques des "gap models".

Tableau a2 (suite).

MODELES	Variables décrivant un arbre						Objectifs ^c							
	Hauteur et/ou diamètre (surface terrière) et/ou volume et/ou biomasse	Variables synthétiques du houppier	Matrice en 3 dimensions			Prévision		Compréhension des mécanismes						
			Houppier simple	Houppier complexe	Houppier réaliste	Production ligneuse Sylviculture (dont climat)	Autres (dont climat)	Composition floristique	Sylviculture Autres (dont climat)	Autres				
Peuplement Type Turnbull (1963) ou Lemoine (1995)	*	-	-	-	-	-	*	-	-	-	-	-	-	-
Peuplement avec distribution <u>Fonctions de distribution.</u> Type Watts (1988), Kuru (1992) Type Kohyama (1991, 1992 a et b) <u>Tables structurées</u> Type FORMIX2 ^b (Bossel et Krieger, 1991) Type Favrichon (1995) <u>Cohortes</u> Type CAFOGROM (Alder, 1994, 1995)	*	-	-	-	-	-	*	-	-	-	*	-	-	-
Individuels indépendants de la distance <u>Empiriques</u> Type Stand Prognosis (Wykoff et al., 1982) et STEMS (Belcher et al., 1982)	*	-	-	-	-	-	*	-	*	-	-	-	-	-
Type DIPSIM (Ong et Kleine, 1996) (à suivre)	*	-	-	-	-	-	*	-	-	*	-	-	-	-

Tableau a2 (fin).

MODELES	Variables décrivant un arbre						Objectifs ^c						
	Hauteur et/ou diamètre (surface terrière) et/ou volume et/ou biomasse	Variables synthétiques du houppier	Matrice en 3 dimensions			Prévision		Compréhension des mécanismes					
			Houppier simple	Houppier complexe	Houppier réaliste	Production ligneuse Sylviculture	Autres (dont climat)	Sylviculture	Autres (dont climat)	Composition floristique	Autres (dont climat)		
(suite) Fonctionnels "Gap models"	*	-	-	-	-	-	*	*	*	*	*	*	*
Individuels dépendants de la distance <u>Empiriques</u> Type Mitchell (1969, 1975) Type FOREST (Ek et Monserud, 1974a) Type Hubbell et al. (1990) Type SILVA (Pretzsch, 1992a et b) Type Courbaud (1991, 1995)	*	* (détaillé)	-	-	-	-	*	*	*	*	*	*	*
SELVA (notre modèle)	*	-	-	-	-	-	*	-	-	-	-	-	-
<u>Fonctionnels</u> Type SORTIE (Pacala et al., (1993) Type SPGROW (Mou et al., 1993) Type ARCADIA (Williams, 1996)	*	-	-	-	-	-	-	-	-	-	*	*	*
Modèles architecturaux Type AMAPpara (Blaise et de Reffye, 1994)	*	-	-	-	-	*	-	-	-	-	-	-	-

^b D'après le classement de Vanclay, 1994, mais FORMIX possède de nombreuses caractéristiques des "gap models".

^c Il s'agit des objectifs principaux affichés par les auteurs. Certains modèles ont cependant des potentialités plus larges.

Tableau a3 (9 pages). Exemples de modèles de croissance. Dans tous les cas, la variable D désigne le diamètre à 1m30, C la circonférence et H la hauteur totale de chaque individu. ΔD , ΔC et ΔH sont les accroissements correspondants. Les b_i sont des paramètres estimés. Les coefficients de détermination R^2 reportés dans la colonne « Efficacité » se rapportent à une validation sur les accroissements individuels (et non sur des variables globales de peuplement).

TYPE DE MODELE	AUTEURS (nom du modèle)	EXPRESSION	Type de peuplement	Efficacité annoncée (coefficients de détermination R^2)
Modèles polynômiaux				
<i>Indépendant de la distance</i>				
Empirique ^a (dendrométrique)	Wykoff et al. (1982) (Stand Prognosis)	$\text{Log}(dds) = \text{HAB} + \text{LOC} + b_1 \cos(\text{ASP}) \text{SL} + b_2 \sin(\text{ASP}) \text{SL} + b_3 \text{SL} + b_4 \text{SL}^2 + b_5 \text{EL} + b_6 \text{EL}^2 + b_7 \text{CCF}/100 + b_8 \text{L-og}(D) + b_9 \text{CR} + b_{10} \text{CR}^2 + b_{11} \text{BAL}/100 + b_{12} D^2$ <p>dds = accroissement de (diamètre sous écorce)², sur 10 ans (directement proportionnel à l'accroissement en surface terrière).</p> <p>HAB et LOC codent pour différents types d'habitat et de localisation géographique, ASP et SL caractérisent la pente (orientation et inclinaison), EL = altitude, CCF = « Crown Competition Factor » (Krajicek et al., 1961), indice de compétition indépendant de la distance, basé sur la dimension des houppiers et qui estime « the area available to the average tree in the stand in relation to the maximum area it could use, if it were open-grown », CR = « crown ratio », rapport entre longueur du houppier et hauteur totale de l'arbre, BAL = surface terrière cumulée des individus de diamètre supérieur à celui du sujet.</p> <p>L'accroissement en hauteur est calculé par le même type de modèle, incluant ou non la variable D en fonction de la taille des individus.</p>	Peuplements mélangés inéquiennement tempérés (Etats-Unis)	Non précisé dans le document cité.
	Wan Razali (1986) pour le groupe des Diptérocarpacées. Le même type d'équation est fourni pour les non Diptérocarpacées (non repris ici)	<p>1) $\Delta D = b_0 + b_1 D + b_2 D^2 + b_3 \text{RS} + b_4 \text{GRBA} + b_5 D^* \text{RS} + b_6 D^* \text{GRBA}$ avec RS = surface terrière du groupe Diptérocarpacées ramenée à la surface terrière totale (de la placette) et GRBA = surface terrière du groupe Diptérocarpacées sur la placette.</p> <p>2) $\Delta D = b_0 + b_1 D + b_2 D^2 + b_3 \text{LDG} + b_4 \text{BAT} + b_5 \text{LBAG} + b_6 \text{LTBAG} + b_7 D/\text{RS}$ avec LDG = accroissement annuel en diamètre de l'individu durant la période précédente, LBAG = même chose en ce qui concerne la surface terrière de l'individu, LTBAG = même chose en ce qui concerne la surface terrière de tous les individus, BAT = surface terrière totale sur la placette.</p>	<p>Groupe des Diptérocarpacées, en peuplement naturel tropical mixte (Malaisie Péinsulaire)</p> <p>1) $R^2 = 0,20$ 2) $R^2 = 0,49$ (sur le jeu de données d'ajustement).</p>	
	(à suivre)			

Vanclay (1991a)	<p>Forêts denses tropicales humides (Queenstand)</p> $\text{Log}(\Delta D + a) = b_0 + b_1 D + b_2 \text{Log}(D) + b_3 \text{Log}(D) \text{SQ} + b_4 \text{Log}(BA) + b_5 \text{OBA} + b_6 \text{PS} + b_7 \text{TST} e^{-\text{TST}/5}$ <p>avec $a = 0,02$ pour ΔD exprimé en cm/an (tient compte des accroissements négatifs), SQ = indice de qualité du site, BA = surface terrière cumulée sur la placette, OBA = surface terrière cumulée des individus de diamètre supérieur à celui du sujet, PS = variable booléenne tenant compte de l'adéquation de l'espèce à son substrat, TST = temps écoulé depuis un traitement sylvicole.</p>	$R^2 = 0,45$ (sur le jeu de données d'ajustement).
Lejeune (1996)	<p>Hêtre (<i>Fagus sylvatica</i>), en peuplements mélangés inéquiennes dans lesquels il est dominant (Belgique)</p> $\Delta C = b_0 + b_1 C^2 + b_2 (1/C^3) + b_3 \text{g sup} + b_4 \text{GHA} + b_5 (\text{VAR/CMOY}) + b_6 \text{PVEG} + b_7 \text{SABL} + b_8 \text{VS} + b_9 \text{AC} + b_{10} \text{HA}$ <p>avec : g sup = surface terrière cumulée des individus de taille supérieure à celle du sujet, GHA = surface terrière totale sur la placette, VAR = variance des circonférences sur la placette, CMOY = circonférence moyenne quadratique, PVEG = durée de la période de végétation, SABL = variable de texture, VS, AC et HA = variables désignant des secteurs écologiques.</p>	$R^2 = 0,40$ (sur le jeu de données d'ajustement).
Monserud et Sierba (1996)	<p>Forêts mélangées inéquiennes tempérées (Autriche)</p> $\text{Log}(\text{BAI}) = b_0 + b_1 \text{Log}(D) + b_2 D^2 + b_3 \text{Log}(\text{CR}) + b_4 \text{BAL} + b_5 \text{CCF} + b_6 \text{SITE}$ <p>avec : BAI = accroissement en surface terrière sur 5 ans, CR = « crown ratio », rapport entre longueur du houppier et hauteur totale de l'individu, BAL = surface terrière cumulée des individus de diamètre supérieur à celui du sujet, CCF = « Crown Competition Factor » de Krajicek et al. (1961) et SITE = variable décomposée par les auteurs en une série de fonctions dépendant des caractéristiques de pente, d'orientation et d'altitude, de caractéristiques pédologiques, des types de végétation et de la localisation géographique.</p>	$R^2 = 0,20$ à $0,63$ en fonction des espèces (sur le jeu de données d'ajustement).
Ong et Kleine (1996) (DIPSIM)	<p>Peuplements naturels mixtes à Diptérocarpacées (Sabah)</p> $\text{Log}(\text{bat}) = b_0 + b_1 \text{Log}(\text{ba}) + b_2 \text{ba} + b_3 \text{SQ} + b_4 \text{Log}(\text{BA}) + b_5 \text{OTBA}$ <p>avec : bai = accroissement individuel en surface terrière, ba = surface terrière de l'individu, SQ = qualité du site (booléen), BA = surface terrière totale sur la placette, OTBA = surface terrière cumulée des individus de diamètre supérieur à celui du sujet.</p>	$R^2 = 0,33$ à $0,71$ en fonction des groupes d'espèces réalisés (sur le jeu de données d'ajustement).
Dépendant de la distance		
Empirique ^b (dendrométrique)	<p>Bella (1971) + nombreux exemples détaillés dans Gourlet-Fleury (1992a)</p>	$R^2 = 0,50$ à $0,83$ en fonction de l'espèce et de l'âge (sur les jeux de données d'ajustement).
(à suivre)	<p>« Competitive Influence Zone Overlap », indice de compétition de type chevauchement. $CIO = \frac{1}{A_i} * \sum_{j=1}^{nc} a_{ij} * \left[\frac{D_j}{D_i} \right]^{EX}$, avec A_i superficie de la zone d'influence de l'arbre sujet i (zone de rayon R_i fonction du diamètre D_i en situation de croissance libre), et a_{ij} chevauchement avec la zone d'influence du $j^{\text{ème}}$ voisin. EX varie de 1,2 à 2 en fonction des espèces considérées.</p>	

Tableau a3 (suite).

Modèles sigmoïdes simples	
<p><i>Indépendant de la distance, dendrométrique^c</i></p> <p>Paresol (1995)</p>	$PAI = b_0 \left[BA^{b_1} - \frac{BA}{A(1-b_2)} \right]$ <p>avec PAI = accroissement annuel en surface terrière, BA = surface terrière de l'individu et A = valeur maximale asymptotique de la surface terrière individuelle.</p> <p style="text-align: center;">Modèles « potentiel * réducteur » (POT*RED)</p>
<p><i>Indépendant de la distance</i></p> <p>Empirique^d (dendrométrique)</p> <p>Belcher et al. (1982) (STEMS)</p>	<p>AD=POT*RED. $POT = b_0 + b_1 D^{b_2} + b_3 SI * CR * D^{b_4}$, $RED = 1 - e^{-b_0 \left[\frac{BA_{max} - BA}{BA} \right]^{0.5}}$</p> <p>avec SI = indice de fertilité du site, CR = fonction de codage du « crown ratio », rapport entre la longueur du houppier et la hauteur totale de l'arbre (fonction de la surface terrière sur la placette et du diamètre de l'individu), BA_{max} = surface terrière maximale possible pour l'espèce considérée sur la placette, BA = surface terrière sur la placette, $B_0 = f(R)*g(AD)$ et R = rapport entre le diamètre de l'arbre et le diamètre moyen sur la placette, AD = diamètre moyen sur la placette.</p>
<p>Fairweather (1988)</p>	<p>BAG=POT*RED (accroissement moyen annuel en surface terrière). $POT = ((b_0 TBA^{b_1}) - b_2 TBA) * (b_3 + b_4 X_1 + b_5 X_2)$ $RED = b_6 * \left[1 - \left(e^{-\left(\frac{b_7}{BAL} + b_8 D^2 \right)} * \sqrt{BATERM} \right) \right]$</p> <p>avec TBA = surface terrière de l'individu, X_1 et X_2 variables booléennes de type pédologique, BAL = surface terrière cumulée des individus de diamètre supérieur à celui du sujet, BATERM = 1-(BAA/250) et BAA = surface terrière totale sur la placette.</p>
<p>(à suivre)</p>	<p>Peuplements mêlés inéquiennes (Etats-Unis)</p> <p>Peuplements mêlés inéquiennes tempérés (Etats-Unis)</p> <p>Peuplements mêlés inéquiennes (Etats-Unis)</p> <p>$R^2 = 0,12$ à $0,64$ en fonction des espèces, sur le jeu de données d'ajustement.</p> <p>Non précisé dans le document cité.</p> <p>$R^2 = 0,48$ à $0,61$ en fonction des espèces (sur un jeu de données de validation).</p>

Tableau a3 (suite).

Farr et Johnson (1988)	<p>$\Delta H = \text{POT} * \text{RED}$ (accroissement en hauteur sur 10 ans). POT (non fourni explicitement) est dérivé de courbes hauteur=f(indice de fertilité du site, âge).</p> <p>1) $RED = b_0(1 - e^{-b_1 CR^{b_2}})$ 2) $RED = e^{-b_0 BA}$ 3) $RED = b_0(1 - e^{-b_1 CR}) * e^{-b_2 (RELHT^{b_3} - 1)}$</p> <p>avec CR = « crown ratio », rapport entre longueur du houppier et hauteur totale de l'arbre, BA = surface terrière totale sur la placette, RELHT = rapport entre la hauteur totale de l'arbre et la hauteur moyenne des 100 individus les plus gros appartenant à la même espèce (BA et RELHT sont ramenés à l'hectare).</p>	<p>Peuplements équétinés à <i>Tsuga heterophylla</i> : 3) $R^2 = 0,22$ 4) $R^2 = 0,20$ 5) $R^2 = 0,27$. <i>Picea sitchensis</i> : 3) $R^2 = 0,43$ 4) $R^2 = 0,37$ 5) $R^2 = 0,48$ (sur le jeu de données d'ajustement).</p> <p><u>Remarque</u> : dans cette étude, des modèles polynômiaux se sont révélés nettement plus efficaces.</p>
Hilt et Teck (1988)	<p>BAG = POT * RED (accroissement moyen annuel en surface terrière). $POT = b_0 SI(1 - e^{-b_2 D})$ $RED = e^{-b_3 BAL}$</p> <p>avec SI = indice de fertilité du site et BAL = surface terrière cumulée des individus de taille supérieure à celle du sujet.</p>	<p>Peuplements mélangés inéquiennes (Etats-Unis)</p> <p>Non précisé dans l'article cité. Sur le jeu de validation, les prédictions sont cependant systématiquement sous-estimées par le modèle, quel que soit le groupe d'espèces pris en compte.</p>
Parresol (1995)	<p>PAI = POT * RED (accroissement moyen annuel en surface terrière), avec ajustement direct de l'ensemble du modèle aux données.</p> $POT = b_0 \left[BA^{b_1} - \frac{BA}{A^{(1-b_2)}} \right]$ <p>1) $RED = e^{-b_2 CC}$ $(b_2 + b_3 I_1 + b_4 I_2) * \frac{INCL}{100}$ 2) $RED = e$</p> <p>avec : BA = surface terrière de l'individu, A = valeur maximale asymptotique de la surface terrière individuelle, CC = classe de houppier (non explicite), I_1 et I_2 = variables booléennes de nature topographique, INCL = valeur de la pente en degrés.</p>	<p>Forêts tropicales humides à <i>Dacryodes excelsa</i> (« tabonuco forests », Porto-Rico)</p> <p>1) $R^2 = 0,34$ à $0,69$ en fonction des espèces. 2) $R^2 = 0,13$ à $0,51$ en fonction des espèces. (Sur le jeu de données d'ajustement).</p>

(à suivre)

Tableau a3 (suite).

Murphy et Shelton (1996)	$\Delta B = POT * RED$ (accroissement moyen annuel en surface terrière), avec ajustement direct de l'ensemble du modèle aux données. $POT = b_0(1 - e^{b_1 B_t})$	Peuplements inéquiennes de <i>Pinus taeda</i> (Etats-Unis)	$R^2=0,69$ (sur le jeu de données d'ajustement).
	$RED = \frac{1}{1 + e^{(b_2 B_t + b_3 B_s + b_4 D_q)}}$		
	avec B_1 = surface terrière de l'individu, B_s = surface terrière cumulée des individus de taille supérieure ou égale à celle du sujet, B_3 = surface terrière totale sur la placette et D_q = diamètre quadratique moyen sur la placette.		
Fonctionnel (« gap model ») Shugart (1984) (différents « gap models »)	$\Delta D = POT * RED_1 * RED_2 * RED_3$ $POT = \frac{GD \left[1 - \frac{DH}{D_{max} H_{max}} \right]}{274 + 3b_0 D - 4b_1 D^2}$ $RED_1 = c_1(1 - e^{c_2[\varrho(h)-c_3]})$: prise en compte de l'effet « lumière », $RED_2 = \frac{4(D_{40} - D_{min 40}) * (D_{max 40} - D_{40})}{(D_{min 40} - D_{max 40})^2}$: prise en compte de l'effet « température », $RED_3 = 1 - \frac{BAR}{SOILQ}$: prise en compte de l'effet « nutriments » dans JABOWA (Botkin et al., 1972) ou dans KIAMBGRAM (Shugart et al., 1980) ; cette fonction peut également être : $RED_3 = b_2 + b_3 RNA + b_4 RNA^2$ (utilisée dans FORNUT de Weinstein et al., 1982, cités par Shugart) ; avec : G = facteur de croissance que l'on peut calculer à partir des valeurs maximales asymptotiques de la hauteur (H_{max}), du diamètre (D_{max}) et de l'âge (AGEMAX) d'un individu. Q(h) représente la quantité de lumière disponible pour le sujet étant donnée sa hauteur h (loi de Beer) ; c_1 , c_2 et c_3 sont des paramètres caractéristiques de la courbe de réponse de la photosynthèse à la lumière. D_{40} , $D_{min 40}$ et $D_{max 40}$ représentent le nombre de degrés-jour annuels resp. nécessaires à la croissance de l'espèce considérée, minima et maxima supportables par l'espèce (= limites de l'aire de répartition). BAR = surface terrière cumulée sur la trouée, SOILQ = quantité maximale de biomasse forestière supportable par le site, RNA = disponibilité relative en nutriments.	Tous types de peuplements La validation est généralement faite de manière qualitative et à moyen/long terme.	

(à suivre)

Tableau a3 (suite).

Dépendant de la distance,
empirique (dendrométrique)¹

<p>Ek et Monserud (1974b), Monserud (1975) (FOREST)</p>	<p>1) $\Delta H = POT * RED$ (accroissement moyen annuel en hauteur). $POT = f(S, A + LGP) - f(S, A)$ avec $f =$ « site index curve » spécifique, $S =$ estimation de l'indice de fertilité de site, $A =$ âge « physiologique » = $f^1(S, H)$. $RED = OVS * ALLOM * HSHIFT$ et</p>	<p>Peuplements mélangés irréguliers (Etats-Unis)</p>	<p>1) Croissance en hauteur : $R^2 = 0,55$ à $0,85$ en fonction des espèces. 2) Croissance en diamètre : $R^2 = 0,56$ à $0,86$ en fonction des espèces. Sur le jeu de données d'ajustement (dans Monserud, 1975).</p>
---	--	--	---

$OVS = \left[1 - e^{-\frac{-1}{b_0 CIA}} \right]^{b_1}$ dans lequel CIA est un indice de compétition « ajusté » pour tenir compte de la tolérance à l'ombrage des espèces, et de l'effet positif ou négatif d'une intervention sylvicole autour de l'arbre sujet (fonctions entre crochets):

$$CIA = CI * \left[1 - b_2^{(H+1)} \right] * \left[1 + b_3 \Delta CI \right]^{b_4} \text{ avec } CI = \left[\sum_{j=1}^N \left(\frac{O_{ij} * S_j}{A_i * S_i} \right) \right]$$

$N =$ nombre de voisins j en compétition avec i , $A_i =$ zone d'influence du sujet i , de rayon égal à celui de la zone d'un individu de même hauteur qui aurait poussé en situation de croissance libre, $O_{ij} =$ superficie du chevauchement entre les zones d'influence du sujet et de son $j^{\text{ème}}$ voisin, S_i et S_j désignent les tailles relatives des individus : hauteur * rayon du houppier si l'arbre avait poussé en situation de croissance libre ;

$ALLOM = CR^{b_5}$ avec $CR =$ « crown ratio » (rapport entre longueur du houppier et hauteur totale de l'arbre). Ce facteur rend compte de la capacité de l'individu à utiliser les ressources disponibles ;

$HSHIFT = 1 - \left[b_6 \left(1 - b_7^{(H+1)} \right) \right]^{CIA}$ qui rend compte du fait que la croissance en hauteur n'est pas maximale en situation de croissance libre.

2) $DD = POT * RED$
(accroissement moyen annuel en diamètre).

$POT = g(H + \Delta H) - g(H)$,

avec $g =$ fonction spécifique d'allométrie hauteur/diamètre en situation de croissance libre ;
 $RED = OVS * ALLOM$ (voir ci-dessus).

(à suivre)

Tableau a3 (suite).

	Peuplements équiennes de <i>Pinus</i> <i>resinosa</i> (Etats-Unis)	Accroissement en diamètre : 1) $R^2 = 0,51$ 2) $R^2 = 0,51$. Accroissement en hauteur : 1) $R^2 = 0,09$ 2) $R^2 = 0,05$. (Sur le jeu de données d'ajustement).
Martin et Ek (1984)	<p>AD = POT*RED. $POT = b_0 S^{b_1} D^{b_2} - b_3 D$ $RED = e^{-b_4 C^{b_5}}$</p> <p>Variable prédite : accroissement moyen annuel en hauteur. $POT = b_0 S(1 - e^{-b_1(A+1)})^{b_2} - H$ $RED = e^{-b_3 C^{b_4}}$</p> <p>avec : S = indice de fertilité de site, A = âge « physiologique » de l'arbre, déduit de la courbe « site index » et C = indice de compétition dépendant de la distance.</p> <p>1) $C = \sum_j e_{ij} \left[\frac{D_j / D_i}{(d_{ij} + 1)} \right]$ 2) $C = \sum_j e_{ij} \frac{D_j * O_{ij}}{D_i Z_i}$</p> <p>avec : j = voisin de l'arbre sujet i, d_{ij} = distance entre le sujet et son j^{ème} voisin, O_{ij} = superficie du chevauchement entre les deux zones d'influence, Z_i = zone d'influence de l'arbre sujet, e_{ij} = facteur de correction des effets de bord (cf. Gourlet-Fleury, 1992a, pour plus de détails sur ces indices).</p>	

(à suivre)

Tableau a3 (suite).

<p>Pretzsch (1990 cité par Collinet, 1993a, et 1992a et b) (SILVA)</p>	<p>1) $\Delta H = POT * RED$ (accroissement en hauteur sur 5 ans). $POT = f(H)$ (non explicité), optimum rencontré pour un site donné. $RED = fact_1 * fact_2 * fact_3 * fact_4$ et $fact_1 = B^{b_0}$, prise en compte de l'importance du houppier vivant, $fact_2 = b_1 e^{(b_2 \epsilon + b_3 \epsilon^2)}$, influence de la compétition latérale entre houppiers adjacents, $fact_3 = e^{(b_4 \omega SP + b_5 \omega BE)}$, influence de l'ombrage sur le houppier, $fact_4 = (1 + b_6 \Delta \epsilon + b_7 \Delta \omega)^{b_8}$, influence d'une ouverture latérale et verticale</p>	<p>Forêts mélangées inéquennes à <i>Picea abies</i> et <i>Fagus sylvatica</i> (Allemagne)</p>	<p><i>Picea abies</i> : 1) $R^2 = 0,47$ 2) $R^2 = 0,62$. <i>Fagus sylvatica</i> : 1) $R^2 = 0,35$ 2) $R^2 = 0,94$. Sur le jeu de données d'ajustement (résultats tirés de Pretzsch (draft)).</p>
<p>avec : B = "crown ratio" (rapport entre la longueur du houppier et la hauteur totale de l'arbre), $\epsilon =$ facteur « d'enserrement latéral » du houppier, $w =$ facteur d'ombrage (wSP : dû à l'Épicéa et wBE : dû au Hêtre).</p>	<p>2) $DkD = POT * RED$</p>		
<p>(accroissement latéral du houppier).</p>	<p>$POT = g(H)$ en conditions optima (g non explicitée),</p>		
<p>$RED = fact_1 * fact_2$ et</p>	<p>$fact_1 = e^{(b_0 \omega SP + b_1 \omega BE)} - b_2 \epsilon^{b_3}$, influence combinée de la compétition latérale pour</p>		
<p>l'extension des houppiers et horizontale pour l'accès à la lumière,</p>	<p>$fact_2 = (1 + b_4 \Delta \epsilon + b_5 \Delta \omega)^{b_6}$, influence d'une ouverture latérale et verticale,</p>		
<p>avec les mêmes notations que précédemment.</p>			
<p>Courbaud (1991), Courbaud et al. (1993)</p>	<p>$\Delta H = POT * RED$ (accroissement en hauteur sur 5 ans). $POT = \frac{P_2 P_1^{(1-P_3)}}{1 - P_3} H^{P_3} - \frac{P_2}{1 - P_3} H$ $RED = e^{-b_0 OMB_{tot}^2}$ $OMB_{tot} = \sum_{j=1}^N -\cos(\text{azimut}_{ij}) * \frac{L_{ij}}{D_{ij}}$</p>	<p>Peuplements inéquennes à <i>Picea abies</i> en zone de montagne (France)</p>	<p>Environ 30% de la variabilité interindividuelle « expliquée » par le modèle</p>
	<p>avec : P_1, P_2 et P_3 paramètres du modèle logistique généralisé (Chapman-Richards), $N =$ nombre de voisins j en compétition avec i, $\text{azimut}_{ij} =$ azimut sous lequel chaque concurrent j est vu par le sujet i, $L_{ij} =$ hauteur du voisin j visible dans un demi-cône orienté sud, d'ouverture 90°, centré sur le sujet, $D_{ij} =$ distance horizontale entre le sujet et son $j^{\text{ème}}$ voisin.</p>		

(à suivre)

Tableau a3 (fin).

Moravie (1995)	$\Delta R_j = POT * RED$ (accroissement latéral annuel du j ^{ème} des 8 rayons du houppier considéré comme un octogone).	Fofêt dense tropicale humide (Inde)	Prise en compte, par le modèle, de 20 à 30% de la variabilité individuelle selon l'espèce (jeu de données d'ajustement).
	$POT = r(R_j - b) * \left(1 - \frac{R_j - b}{R_{max} - b}\right)$		
	$RED = \left[\prod_{i=1}^{N_{ij}} \left(\frac{h_{ij}}{h_i} \right) \left(1 - \frac{d_{ij}}{10} \right) \right]^{\alpha} * \left[\frac{ST_a}{N_j} \sum_{k=1}^{N_j} ST_k \right]$		
	avec : r = taux de croissance logistique du rayon j en l'absence de compétition (déterminé empiriquement à partir des données d'accroissement maximal en diamètre), R_{max} = valeur théorique maximale du rayon, b=constante de la relation d'allométrie reliant le rayon du houppier au diamètre D, i désigne un voisin gênant, N_{ij} = nombre de voisins gênants dans la direction j, N_j = nombre total de voisins de l'arbre sujet, d_{ij} = distance entre le sujet a et son voisin i, h = hauteur totale, ST = surface terrière d'un individu.		

^a Voir encore Martin et Ek (1984), Tomé (1988), Wykoff et Monserud (1988), Chai et Lemay (1993).

^b Voir également Beck (1974), Lin (1974), Martin et Ek (1984).

^c Voir également Loffeier (1989) et Durrieu de Madron (1993).

^d Voir également Martin et Ek (1984), Meldahl et al. (1988), Wykoff et Monserud (1988), Wensel et Biging (1988) pour CACTOS, Dentand (1990), Dhôte (1990), Quicke et al. (1994).

^e On pourra se reporter au travail en cours (thèse ENGREF/CIRAD-Forêt) de Picard sur ce type de modèles.

^f Voir aussi Arney (1974), Hegyi (1974), PTAEDA2 de Burckhart et al. (1987, cités par Korzhukin et al., 1996), MOSES de Hasenauer (1994), largement basé sur les équations de FOREST (Monserud, 1975).

Tableau a4 (4 pages). Exemples de modèles de mortalité. P_s et P_m sont respectivement les probabilités de survie et de mort de l'individu au cours du prochain cycle de simulation. D représente le diamètre à 1m30 en début de cycle et ΔD l'accroissement de cette variable durant la période précédente. En cours de simulation, elle est calculée par un modèle de croissance. Les coefficients b_i sont ajustés aux données.

TYPE DE MODELE	AUTEURS (nom du modèle)	TYPE DE MORTALITE	EXPRESSION	Type de peuplement
<i>Indépendant de la distance</i>				
Empirique (dendrométrique)	Wykoff et al. (1982) (Stand Prognosis)	Compétition	$P_m = (1 + e^{(b_0 + b_1 D + b_2 D^2)})^{-1}$ <p>NB : $b_2=0$ la plupart du temps. Deux autres modèles, prenant en compte la densité dans le peuplement, prennent le relai lorsque celle-ci dépasse un certain seuil. Dans les peuplements très denses, l'effet « âge » est introduit pour renforcer la probabilité de mort.</p>	Peuplements mélangés inéquiennes (Etats-Unis)
	Wan Razali (1986)	Compétition	$P_m = (1 + e^{-(b_0 + b_1 D + b_2 D^2 + b_3 \Delta D + b_4 \Delta BA)})^{-1}$ <p>pour le groupe des Diptérocarpacées. Avec : ΔBA = accroissement moyen individuel en surface terrière. Des équations analogues sont fournies pour d'autres groupes d'espèces (les variables explicatives n'étant pas toujours les mêmes).</p>	Forêts naturelles tropicales mixtes (Malaisie Péninsulaire)
	Vancley (1989)	Exploitation	$P_m = (1 + e^{-\left(b_0 + b_1 D + \frac{b_2}{TSL} + b_3 I\right)})^{-1}$ <p>avec P_m = probabilité d'être marqué pour l'exploitation, TSL = temps écoulé depuis la dernière exploitation, $I = CL$ (seuil diamétrique d'exploitabilité, tabulé) ou $I = BA_{merch}$ (surface terrière cumulée des arbres exploitables de plus de 40 cm de diamètre), selon le groupe d'espèces auquel on s'intéresse.</p>	Forêts denses tropicales humides (Queenland, Australie)
		Dégâts d'exploitation	$P_m = (1 + e^{-(b_0 + b_1 RBA + b_3 SLOPE + b_4 D)})^{-1}$ <p>avec P_m = probabilité d'être détruit par l'exploitation, RBA = rapport entre la surface terrière totale éliminée par exploitation et la surface terrière initiale sur la placette, SLOPE = pente en degrés.</p>	
(à suivre)				

Tableau a4 (suite)

Vanclay (1991c)	Compétition	$P_s = (1 + e^{-(b_0 + b_1 \text{Log}(D) + b_2 D + b_3 RS^3 + b_4 SQ + b_5 BA + b_6 \text{Log}(BA))})^{-1}$ <p>la valeur des paramètres variant en fonction des groupes d'espèces. Avec : RS = « statut relatif » de l'arbre sur la placette (position sur la courbe cumulée des surfaces terrières. RS=0 pour l'arbre le plus gros et RS=1 pour le plus petit), SQ = indice de fertilité de site et BA = surface terrière totale sur la placette.</p>	Forêts denses tropicales humides (Queensland, Australie)
Ong et Kleine (1996) (DIPSIM)	Compétition	$P_m = (1 + e^{-(b_0 + b_1 \text{Log}(D) + \frac{b_2}{ht} + b_3 \text{Log}(BA))})^{-1}$ <p>pour les espèces pionnières. Avec : ba = surface terrière de l'arbre sujet et BA = surface terrière totale sur la placette.</p> <p>Pour les autres espèces, $P_m = \alpha(c)$, avec α constante fixée dans chaque classe de diamètre c. Un compteur attaché à chaque arbre permet de gérer les faibles probabilités dans les grandes classes de diamètre.</p> <p>$P_m = \beta(c)$, β étant fixé par l'utilisateur pour deux classes de diamètre : 10-40 cm et ≥ 40 cm. Ce risque est évalué au moment des interventions sylvicoles.</p>	Peuplements naturels mixtes à Diptérocarpacées (Sabah)
Fonctionnel (« gap model »)	Sénescence	$P_m = 1 - e^{-\left(\frac{4,605}{AGEMX}\right)}$ <p>Avec AGEMX = âge auquel 1% des individus d'une cohorte sont encore vivants (tables) et auquel on suppose que leur taille (H et D) a atteint le maximum asymptotique.</p> <p>$P_s = 0,368$ si $\Delta D < 1$ mm/an.</p> <p>$P_m = 0,003$ (chaque année). Les individus subsistant dans la trouée voient leur probabilité de mourir augmenter lorsqu'un chablis s'est produit : $P_m = 0,50$ à $0,75$.</p> <p>$P_m = 0,30$ pour tous les individus de la trouée si une exploitation a eu lieu.</p>	Forêt subtropicale humide (Australie)
	Compétition Chablis		
	Dégâts dûs à l'exploitation		
<i>Dépendant de la distance</i>			
Empirique (dendrométrique) (à suivre)	Compétition	<p>$P_s = 0$ si CR < 0,05. $P_s = 1$ sinon. Avec CR = « crown ratio », rapport entre la longueur du houppier et la hauteur.</p>	Peuplements équinoxiaux monospécifiques de Douglas (<i>Pseudotsuga menziesii</i>) (Etats-Unis)

Tableau a4 (suite)

Hegy (1974)	Compétition	Le nombre d'arbres à éliminer est déterminé à partir d'une comparaison entre l'état du peuplement simulé et tables de production normalisées à un instant donné. 80% des individus à éliminer sont choisis comme étant ceux qui possèdent l'indice de compétition le plus élevé. Les 20% restant sont éliminés au hasard.	Peuplements équiennes monospécifiques de <i>Pinus banksiana</i> (Etats-Unis)
Mitchell (1975)	Compétition	La mortalité dépend de la densité n du peuplement, au travers de l'évaluation d'un seuil critique : $N = \frac{h}{(f * H_s)^2}$ (Wilson, 1951, cité par Mitchell, 1975) avec h et f constantes et H_s hauteur moyenne quadratique des arbres sur la station (calculée sur les 40% d'individus les plus hauts). Quand $n \leq N$, on élimine au hasard 16 arbres dominés/ha. Quand $n > N$, on élimine tous les arbres dominés.	Peuplements équiennes monospécifiques de Douglas (<i>Pseudotsuga menziesii</i>) (Etats-Unis)
Monserud (1975) (FOREST)	Compétition	$P_s = (1 + e^{-(b_0 + b_1 D_{pr,d} + b_2 \Delta D_{pr,d} + b_3 CI)})^{-1} - LGP$ Avec : $D_{pr,d}$ et $\Delta D_{pr,d}$ = respectivement diamètre et accroissement diamétrique prédits par le modèle de croissance présenté au tableau a3, CI = indice de compétition (voir formulation au tableau a3) et LGP = longueur de la période de croissance prise en compte. Efficacité (sur le jeu de données d'ajustement) : classification correcte de 88% des arbres vivants et de 35% des arbres morts.	Peuplements mélangés irréguliers (Etats-Unis)
Belcher et al. (1982) (STEMS)	Compétition	$P_m = (1 + e^{(b_0 + b_1 \Delta D^2 + b_2 D)})^{-1} + b_4$	Peuplements mélangés inéquiennes (Etats-Unis)
Hasenauer (1994)	Compétition	$P_s = (1 + e^{(b_0 + b_1 CIA + b_2 CR + b_3 D)})^{-1}$ Avec : CIA = indice de compétition corrigé de Ek et Monserud (cf. tableau a3), CR = « crown ratio », rapport entre longueur du houppier et hauteur totale de l'arbre, D = diamètre à 1m30 en début de période de calcul.	Peuplements mélangés inéquiennes (Autriche)

(à suivre)

Tableau a4 (fin)

Moravie (1995)	Compétition	$P_m = P_{i,C} - K \frac{\Delta D - \Delta D_{moyi,C}}{\Delta \max i,C}$ <p>et $\Delta \max i,C = \max \left(\left \Delta D_{\max i,C} - \Delta D_{moyi,C} \right , \left \Delta D_{\min i,C} - \Delta D_{moyi,C} \right \right)$.</p> <p>Avec : i et C désignent respectivement l'espèce et la classe de diamètre auxquels appartient l'arbre sujet. $P_{i,C}$ = probabilité moyenne de mort au sein de la classe (i,C). $\Delta D_{\max i,C}$ (resp. $\Delta D_{\min i,C}$ et $\Delta D_{moyi,C}$) = accroissement maximal (resp. minimal et moyen) observé parmi les arbres appartenant à (i, C) pendant la période précédente, K = facteur permettant de faire varier l'influence de la compétition sur la mortalité.</p>	Forêt dense tropicale humide (Inde)
Williams (1996) (ARCADIA)	Compétition	$P_m = 1 - e^{-\left(\frac{b_0 + \frac{b_1}{\left(1 + \frac{\min(V_{sr}, G_r)}{q} \right)^3} \right)}{D_{hc}^2} \cdot G_r = \frac{A_{tot}}{L_{tot} A_{max}}$ <p>Avec : D_{hc} = diamètre à la base du houppier, D_{bt} = diamètre à la base du tronc, V_{sr} chiffre l'aptitude d'un individu à résister aux stress (par l'intermédiaire de son stock d'assimilats), A_{tot} = total des assimilats exportés pendant l'année (différence entre le taux de photosynthèse effectif et les différents coûts liés à la respiration, l'élaboration des feuilles, des racines, la maintenance de la structure), L_{tot} = surface foliaire totale de l'individu, A_{max} = taux d'assimilation maximum par la photosynthèse, G_r chiffre l'efficacité d'utilisation des assimilats pour la croissance et q = valeur seuil.</p> <p>Tout individu mort de plus de 20 cm de diamètre occasionne des dégâts : voir fonction suivante.</p>	Peuplements mélangés inéquiennes tempérés (Etats-Unis)
	Dégâts occasionnés par la chute d'un arbre	$P_m = 1,3 - \frac{V_{ii}}{V_{ik}}$ <p>Avec : V_{ii} = volume de l'arbre qui tombe, V_{ik} = volume de l'arbre sujet. Les sujets sont pris en compte dans les voxels (cellules 3D) situés dans l'axe de chute de tout arbre de $D \geq 20$ cm, axe lité au hasard en l'absence d'informations sur les vents dominants.</p>	
	Ouragans	<p>La probabilité pour que se produise un ouragan, une année donnée, est une constante. Lorsque l'ouragan se produit, chaque arbre se voit attribuer un risque de chute : $P_m = f(D)$. Il faut ensuite ré-évaluer les dégâts potentiellement occasionnés par ces chutes.</p>	

Tableau a5 (4 pages). Exemples de modélisation des processus de régénération. Les coefficients b_i sont ajustés aux données.

TYPE DE MODELE	AUTEURS (nom du modèle)	Processus modélisé	EXPRESSION	Type de peuplement
<i>Indépendant de la distance ^a</i>				
Empirique (dendrométrique)	Vanclay (1989, cité par Vanclay, 1992)	Recrutement	<p>Dans un premier temps, l'auteur prédit le nombre d'individus apparaissant au-dessus de 20 cm de diamètre à partir de deux variables : surface terrière totale cumulée sur la placette et indice de qualité de site. La proportion de recrutés attribuée à chaque groupe d'espèces en présence est ensuite calculée de la manière suivante :</p> $P_i = \frac{1 + e^{(b_0 + b_1 BA + b_2 B_i + b_3 B_i SQ)}}{1 + e^{(b_0 + b_1 BA + b_2 B_i + b_3 B_i SQ)}} \quad \text{sous la contrainte } \sum_i P_i = 1, \quad \text{avec}$ <p>P_i = proportion attribuée au $i^{\text{ème}}$ groupe, BA = surface terrière totale cumulée sur la placette, B_i = surface terrière totale des individus appartenant au $i^{\text{ème}}$ groupe, déjà présents sur la placette, SQ = indice de qualité de site et les b_j = coefficients de régression dépendant des groupes d'espèces.</p>	Forêts denses tropicales humides (Queensland)
	Vanclay (1992)	Recrutement	$P = \frac{1 + e^{-(b_0 + b_1 PRES + b_2 BA + b_3 \text{Log}(BA) + b_4 \text{SOIL}_1 + b_5 TR)}}{1 + e^{-(b_0 + b_1 PRES + b_2 BA + b_3 \text{Log}(BA) + b_4 \text{SOIL}_1 + b_5 TR)}} \quad \text{et}$ $\text{Log}(N) = b_6 + b_7 \text{Log}(BA) + b_8 \text{Log}(RNO + 0,2) + b_9 SQ + b_{10} \text{SOIL}_2 \quad \text{pour 22}$ <p>groupes d'espèces présentant des stratégies de recrutement homogènes ; avec P = probabilité pour un individu appartenant à l'un des groupes pris en compte d'être recruté une année donnée, PRES = variable booléenne indiquant la présence ou l'absence sur la placette du groupe considéré, BA = surface terrière totale sur la placette, SOIL_1 et SOIL_2 = variables booléennes codant pour un type de sol, $TR = t e^{-t}$ = réponse à un traitement sylvicole quelconque (t est le temps écoulé depuis une intervention) assurant une probabilité de recrutement maximale 9 ans plus tard, N = effectif moyen recruté annuellement pour le groupe pris en compte sachant P, RNO = effectif du groupe rapporté à l'effectif total présent sur la placette, SQ = indice de qualité de site et les b_j = coefficients de régression dépendant des groupes d'espèces.</p>	Forêts denses tropicales humides (Queensland)

(à suivre)

Tableau a5 (suite).

Alder (1994) CAFOGROM ^a	Recrutement	<p>Nombre de nouveaux individus à recruter au-dessus de 5 cm de diamètre sur une période de 5 ans :</p> $N = b_0 + b_1 BA, \text{ un effectif minimum de } 50/\text{ha}/5 \text{ ans étant imposé.}$ <p>Répartition de cet effectif entre 3 groupes d'espèces : « pionnières », « tolérantes à l'ombrage » et « intolérantes à l'ombrage » (proportions respectives : P_p, P_{tol}, P_{intol}) :</p> $\text{logit}(P_p) = \text{Log}\left(\frac{P_p}{1 - P_p}\right) = b_2 + b_3 \text{Log}(N),$ $\text{logit}(P_{intol}) = \text{Log}\left(\frac{P_{intol}}{1 - P_{intol}}\right) = b_4 + b_5 \text{Log}(N)$ <p>et $P_{tol} = 100 - (P_p + P_{intol})$,</p> <p>avec BA = surface terrière totale sur la placette, P_p, P_{tol} et P_{intol} = proportions respectives des 3 groupes d'espèces cités plus haut dans l'ensemble des individus recrutés.</p>	Forêts denses tropicales humides du bassin amazonien (Brésil)
Fonctionnel (« gap model »)	Dissémination	<p>Homogène dans l'espace : les 125 espèces considérées sont supposées disposer d'une source de graines permanente.</p> <p>Irrégulière dans le temps : les espèces sont classées en fonction du rythme de production de cette source (irrégulière : on tire au hasard une année sur 3 pour la production ; régulière : des graines sont disponibles chaque année) et de la longévité des graines.</p>	Forêt subtropicale humide (Australie)
	Etablissement de plantules	<p>6 filtres concernant le mode de dispersion des graines, le type de sol, la topographie et la présence de taxons « étrangereurs » permettent de sélectionner les espèces à régénérer. Un générateur de nombres pseudo-aléatoires permet alors de déterminer 1) le nombre total de plants à installer : entre 1 et 20 ou bien 1 et 600 si un chablis s'est produit, par trouée de 1/20 d'ha ; 2) la première espèce de la liste à régénérer, puis le nombre de plants de cette espèce à installer : entre 1 et 20 par trouée ; 3) la deuxième espèce de la liste à régénérer, puis de nouveau le nombre de plants de cette espèce à installer : entre 1 et 20 ; 4) et ainsi de suite jusqu'à atteindre le quota déterminé en 1).</p>	

(à suivre)

Tableau a5 (suite).

Bugmann (1996) (FORCLIM)	4 filtres environnementaux sont utilisés : température hivernale minimale de l'année, seuil minimum de lumière disponible au sol, intensité de l'abrutissement par les cervidés et somme annuelle des degrés-jours non conformes aux exigences de l'espèce considérée. Lorsque tous les filtres sont franchis avec succès, l'auteur calcule pour l'espèce une probabilité d'installation et le nombre de nouveaux individus est tiré au hasard en tenant compte d'une capacité maximum du site (0,006 jeunes plants/m ² /an) et de la taille de la trouée.	Forêts mixtes tempérées européennes (Alpes suisses)
Empirique (dendrométrique)	Fructification	L'importance de la production annuelle de graines est définie en pourcentage d'une production maximale possible, pour chaque espèce, en condition de « fully stocked stand » (valeurs fournies par la littérature) et est fixée par un générateur de nombres pseudo-aléatoires. Elle dépend de la taille (diamètre et hauteur) des individus reproducteurs.
Monserud (1975) (FOREST, Ek et Monserud, 1974a)	Dissémination	Elle est réalisée selon un schéma dans lequel la répartition des graines dans les quadrats est une fonction exponentiellement décroissante de la distance au pied-mère. Des termes aléatoires permettent de gérer la possibilité d'apparition d'espèces particulières en l'absence de reproducteurs sur la parcelle.
	Germination et installation des plantules	Le nombre de plantules installées sur chaque quadrat élémentaire est déduit du stock de graines par application d'un taux de survie puis d'un taux de germination, fixés par un générateur de nombres pseudo-aléatoires à partir de gammes de valeurs trouvées dans la littérature.
	Croissance des juvéniles et recrutement	Les plantules entrent dans la première classe de hauteur d'un ensemble juvénile géré par un modèle de type peuplement tenant à la fois du modèle matriciel et du modèle de cohorte. Le peuplement est constitué de 4 classes de hauteur ; la structure totale est considérée comme stable, ce qui permet de déterminer à chaque cycle les taux de mortalité par classe. Chaque classe est gérée comme une cohorte, c'est-à-dire qu'elle est assimilée à un seul individu, affecté d'un facteur d'expansion (l'effectif de la classe à chaque cycle) et supposé situé au centre de chaque quadrat. Le modèle de croissance en hauteur appliqué à ce super-individu est du type « potentiel*réducteur » et construit de la même manière que le modèle des adultes (cf. tableau a3). Le passage d'une classe à l'autre se déduit du ratio croissance prédite / largeur de la classe originelle. Lorsque l'individu dépasse la limite supérieure de la dernière classe (25 pieds : 7,60 m), on lui affecte des coordonnées dans son quadrat de naissance.

(à suivre)

Tableau a5 (fin).

Fonctionnel	Williams (1996) (ARCADIA)	Floraison, fructification Dissémination	Parvenus à un certain stade de maturité, les adultes peuvent distraire une partie de leurs assimilats pour la genèse de fleurs, de fruits et de graines. Elle est fonction de la hauteur des pieds-mères et d'un « facteur de dispersion » spécifique déterminé selon la méthode de Ribbens et al., 1994 (voir paragraphe 2.3.3.3 dans le texte principal et annexe 2). Chaque cellule tri-dimensionnelle met à jour annuellement sa banque de graines. Elle dépend de différents facteurs : la cellule concernée doit être vide de tout tronc et de tout feuillage, de même que la cellule qui la surplombe. Un certain degré d'éclairement (à peu près 5% de l'éclairement total incident sur la canopée), des conditions de température minima et maxima, des conditions portant sur la sécheresse annuelle sont par ailleurs requis et permettent de déterminer le nombre de graines viables au bout d'un an. La probabilité pour qu'une graine fournisse une plantule dépend de son poids. Elle est calculée pour chaque espèce en présence, puis normalisée pour tenir compte du fait qu'un seul individu pourra s'installer dans la cellule. Un générateur de nombres pseudo-aléatoires détermine alors l'espèce « gagnante » et un nouvel individu faisant entre 1 et 2 cm de diamètre, dont la base du houppier est fixée à 0,5 m de hauteur, est établi.	Peuplements mélangés inéquiennes (Etats-Unis)
		Germination		
		Etablissement des plantules		

^a Il s'agit de modèles individuels, sauf en ce qui concerne CAFOGROM (Alder, 1994), modèle de peuplement avec distribution en cohortes, mais gérant des « super-individus » selon une phytosophie analogue à celle des vrais modèles individuels.

Tableau a6 (2 pages). Exemples d'évaluation « quantitative » du fonctionnement global de quelques modèles individuels de dynamique forestière. L'évaluation est basée sur la comparaison entre valeurs prédites et valeurs observées de diverses origines.

Auteurs (nom du modèle)	Variables prédites et observées	Origine des données observées	Type d'évaluation
<i>Modèles dendrométriques</i>			
Hegyí, 1974	Surface terrière totale, effectif total, diamètre moyen, hauteur moyenne, volume total (une espèce en peuplement pur)	Données expérimentales	Quantitative (calcul des écarts sur les volumes) + comparaison « qualitative » sur 100 ans de développement
Lin, 1974	Surface terrière totale, effectif total, diamètre moyen (deux espèces, chacune en peuplement pur)	Données expérimentales	Comparaison « qualitative » de l'évolution au cours du temps (jusqu'à 27 ans)
Monserud, 1975 (FOREST)	Surface terrière totale, effectif total, distributions de la surface terrière et de l'effectif total par classes de diamètre	Données expérimentales	Quantitative « commentée » (calcul des écarts) de l'évolution en fonction du temps (13 à 26 ans)
Belcher et al., 1982 (STEMS)	Effectif, ST, diamètre moyen (globaux et/ou par espèce)	Autres dispositifs de suivi	Quantitative « commentée » : calcul du « ratio of predicted mean to observed mean », en fonction du temps passé (5 à 17 ans)
Fairweather, 1988	Surface terrière par espèce	Placettes d'inventaire	Comparaison « qualitative » de l'évolution au cours du temps (jusqu'à 14 ans)
Tomé, 1988 (SICREU)	Surface terrière totale, hauteur dominante (une espèce en peuplement pur)	Données expérimentales	Comparaison « qualitative » de l'évolution en fonction du temps (jusqu'à 10 ans)
Siekierski, 1991	Surface terrière totale, effectif total, diamètre moyen (deux espèces en forêt mixte inéquienne)	Données expérimentales	Quantitative : calcul des écarts, test F sur les coefficients de la droite de régression données observées = f(données prédites)
Ottorini, 1995 (SimCoP)	Effectif total, distribution des effectifs par classes de diamètre, volume total sur pied, (une espèce en peuplement pur)	Données expérimentales	Comparaison « qualitative » de l'évolution au cours du temps (jusqu'à 40 ans)
Ong et Kleine, 1996 (DIPSIM)	Distribution des effectifs totaux par classes de diamètre Volume par groupe d'espèces	Etat initial du peuplement Autres dispositifs de suivi	Comparaison « qualitative » Quantitative « commentée » : calcul des pseudo-R ² , pour différentes périodes de projection (10 ou 20 ans)

(à suivre)

Tableau a6 (fin).

<i>Modèles fonctionnels</i>			
Doyle, 1981, cité par Shugart, 1984 (FORICO)	Distribution des effectifs totaux par classe de diamètre	Autres dispositifs de suivi	Quantitative (calcul des écarts) + test de Kolmogorov
Mou et al., 1993 (SPGROW)	Biomasse totale (3 espèces différentes, chacune en peuplement pur)	Données expérimentales	Quantitative (calcul des écarts) + comparaison qualitative d'histogrammes, au bout d'un an
Pacala et al., 1993, 1996 (SORTIE)	Surface terrière globale, surface terrière de chaque espèce	Littérature concernant différentes expérimentations dans le même type de forêt, et concernant la composition floristique de forêts matures (« old- growth) pour évaluer les prédictions à long terme	Comparaison « qualitative » de l'évolution au cours du temps (jusqu'à 2000 ans)
Williams, 1996 (ARCADIA)	Surface terrière totale, biomasse totale, indice foliaire global	Littérature	Comparaison « qualitative » de valeurs à l'équilibre (au- delà de 1000 ans)

Tableau a7. Structure, en 1988, du peuplement étudié sur les huit parcelles d'étude.

Classes de diamètre	Peuplement total*		Peuplement potentiel**		Peuplement étudié		Code .4	
	Eff. abs.	Eff./ha	Eff. abs.	Eff./ha	Eff. abs.	Eff./ha	Eff. abs.	Eff./ha
10-15	9752	195,04	4977	99,54	4197	83,94	299	5,98
15-20	5740	114,80	3161	63,22	2407	48,14	304	6,08
20-25	3597	71,94	1929	38,58	1436	28,72	241	4,82
25-30	2222	44,44	1202	24,04	920	18,40	162	3,24
30-35	1604	32,08	859	17,18	657	13,14	117	2,34
35-40	1257	25,14	711	14,22	517	10,34	129	2,58
40-45	616	12,32	335	6,70	250	5,00	59	1,18
45-50	463	9,26	225	4,50	150	3,00	55	1,10
50-55	240	4,80	120	2,40	73	1,46	43	0,86
55-60	161	3,22	89	1,78	61	1,22	26	0,52
60-65	110	2,20	52	1,04	37	0,74	15	0,30
65-70	62	1,24	31	0,62	20	0,40	10	0,20
70-75	36	0,72	13	0,26	5	0,10	7	0,14
75-80	32	0,64	18	0,36	7	0,14	10	0,20
80-85	19	0,38	9	0,18	3	0,06	6	0,12
85-90	17	0,34	5	0,10	1	0,02	3	0,06
90-95	9	0,18	4	0,08	3	0,06	1	0,02
95-100	7	0,14	1	0,02	0	0,00	1	0,02
100-105	5	0,10	2	0,04	0	0,00	1	0,02
105-110	5	0,10	2	0,04	0	0,00	1	0,02
≥110	10	0,20	4	0,08	0	0,00	4	0,08
Total	25964	519,28	13749	274,98	10745	214,90	1494	29,88
				52,95	78,15			10,87

* Ensemble des arbres (toutes espèces) vivants sur les huit parcelles en 1988.

** Ensemble des arbres situés à plus de 30 m des bords des parcelles et vivants entre 1985 et 1991.

Tableau a8 (6 pages). Quelques caractéristiques des 173 espèces représentées dans le jeu de données d'ajustement. Diamètre moyen et 3ème quartile concernent leur distribution diamétrique sur l'ensemble des 12 parcelles à la campagne 5 (1988). Moyenne et variance résiduelles sont calculées sur les résidus du modèle [13] après élimination de l'effet parcelle par le biais d'une ANOVA à un facteur. R^2 : coefficient de détermination de la régression $\text{Log}(\Delta D+0,287) = a\text{NBD}+b\text{ANBD}+c$. Sans indication : $p \leq 0,001$ (NS : coefficient non significativement différent de 0 ; * : $0,01 \leq p < 0,05$; ** : $0,001 \leq p < 0,01$). Le coefficient n'a été calculé que pour les espèces représentées par plus de 10 individus. Les lignes en gras correspondent aux 87 espèces qui ont servi à la "fabrication" des groupes.

Espèce	Code	Effectif	Diam. moyen (cm)	3ème quartile (cm)	R2	Moy. résiduelle (Log(cm))	Var. des résidus (Log(cm)²)	Groupe
<i>Eschweilera sagotiana</i> et spp (LECY)	403	1079	23,32	29,60	0,16	-0,11	0,11	g2.2
<i>Lecythis corrugata</i> et spp (LECY)	404	830	16,92	19,74	0,08	-0,20	0,08	g1.1
<i>Licania</i> spp (CHRY)	401	542	19,85	24,51	0,22	-0,04	0,08	g1.3
<i>Eperua falcata</i> (CAES)	224	431	28,72	38,20	0,18	0,08	0,13	g2.4
<i>Licania</i> cf. <i>micrantha</i> (CHRY)	705	398	23,33	29,92	0,25	-0,02	0,07	g2.3
<i>Sloanea</i> cf. <i>guianensis</i> (ELAE)	703	385	15,36	17,63	0,18	-0,01	0,06	g1.4
<i>Oxandra asbeckii</i> et spp (ANNO)	614	336	12,64	13,69	0,19	-0,11	0,05	g1.2
<i>Licania</i> <i>alba</i> (CHRY)	701	299	22,27	27,50	0,16	-0,07	0,08	g2.3
<i>Couepia</i> cf. <i>caryophylloides</i> (CHRY)	704	253	18,98	23,08	0,35	0,16	0,09	g5.2
<i>Iryanthera sagotiana</i> (MYRI)	222	246	18,31	22,12	0,24	-0,12	0,10	g1.2
<i>Bocopa prouacensis</i> (CAES)	105	245	23,10	29,60	0,14	-0,19	0,07	g2.1
<i>Symphonia globulifera</i> (CLUS)	607	228	21,92	27,69	0,29	0,35	0,11	g4.2
<i>Tovomita</i> sp2 (CLUS)	616	191	14,57	16,55	0,23	-0,06	0,07	g1.4
<i>Micropholis guianensis</i> (SAPO)	629	190	21,16	24,75	0,37	0,08	0,11	g5.2
<i>Eperua grandiflora</i> (CAES)	698	167	27,12	36,61	0,28	0,29	0,11	g3.1
<i>Carapa procera</i> (MELI)	207	159	21,46	27,50	0,36	0,25	0,11	g4.2
<i>Protium</i> sp (BURS)	314	158	19,39	23,94	0,32	0,21	0,11	g5.3
<i>Pradosia cochlearia</i> (SAPO)	402	155	32,28	42,97	0,07**	-0,02	0,15	g2.3
<i>Licania</i> cf. <i>hypoleuca</i> (CHRY)	709	149	21,34	27,06	0,22	0,07	0,10	g2.4
<i>Gustavia hexapetala</i> et <i>G. augusta</i> (LECY)	602	148	14,27	16,31	0,11	-0,13	0,06	g1.2
<i>Dicorynia guianensis</i> (CAES)	103	138	28,54	38,67	0,29	0,23	0,13	g3.1
<i>Simaba cedron</i> (SIMA)	618	122	13,03	14,32	0,18	-0,23	0,06	g1.1
<i>Lecythis poiteauii</i> (LECY)	633	118	21,93	27,50	0,15	-0,10	0,09	g2.2
<i>Qualea rosea</i> (VOCH)	604	111	35,66	49,34	0,31	0,19	0,13	g3.1
<i>Mouriri</i> cf. <i>collocarpa</i> (MELA)	662	110	17,63	21,65	0,38	0,06	0,05	g5.1
<i>Couratari multiflora</i> et spp (LECY)	308	109	22,33	29,13	0,33	-0,01	0,10	g2.3

Talisia spp (SAPI)	655	96	16,42	18,30	0,32	0,00	0,07	g5.1
Sterculia sp (STER)	310	93	23,62	30,88	0,35	0,28	0,25	g4.2
Recordoxylon speciosum (CAES)	117	84	26,16	36,53	0,33	0,12	0,11	g4.1
Pouteria guianensis (SAPO)	606	84	20,04	23,87	0,42	-0,03	0,07	g5.1
Pouteria melanopoda (SAPO)	643	84	24,85	29,92	0,24	0,02	0,09	g2.3
Sagotia racemosa (EUPH)	687	84	12,00	12,54	0,21	-0,02	0,03	g1.4
Drypetes variabilis (EUPH)	707	84	19,72	23,08	0,39	0,00	0,07	g5.1
Chrysophyllum sanguinolentum (SAPO)	206	78	26,83	35,17	0,42	0,13	0,10	g4.1
Chaetocarpus schomburgkianus (EUPH)	617	72	24,34	29,28	0,17**	0,01	0,09	g2.3
Posoqueria latifolia (RUBI)	627	72	16,20	18,78	0,16**	-0,07	0,07	g1.4
Catostemma fragrans (BOMB)	641	72	16,69	19,42	0,11*	-0,17	0,06	g1.1
Trymatococcus oligandrus (MORA)	679	69	17,64	20,77	NS	-0,11	0,11	g1.2
Moronobea coccinea CLUS)	608	67	25,35	33,90	0,20	0,18	0,13	g2.4
Eriotheca globosa et sp (BOMB)	312	63	18,43	21,01	0,51	0,03	0,15	g5.1
Anaxagorea dolichocarpa (ANNO)	644	60	12,23	13,21	0,17**	-0,18	0,05	g1.1
Theobroma subincanum (STER)	623	57	17,38	20,37	0,29	-0,07	0,08	g1.4
Lacmellea floribunda (APOC)	313	55	14,46	16,07	0,23**	-0,07	0,07	g1.4
Tapura sp (DICH)	659	54	23,50	27,50	0,38	-0,09	0,05	g2.2
Chrysophyllum prieurii (SAPO)	619	49	25,73	32,79	0,21**	0,10	0,09	g2.4
Sloanea cf. grandiflora (ELAE)	653	48	19,29	23,40	0,50	-0,01	0,08	g5.1
Ocotea rubra (LAUR)	214	47	32,20	43,93	0,38	0,21	0,13	g3.1
Virola michelii (MYRI)	658	47	23,53	32,60	0,39	0,34	0,10	g4.2
Goupia glabra (CELA)	212	43	32,60	47,27	NS	0,12	0,15	g2.4
Conceveiba guianensis (EUPH)	601	42	16,18	19,74	0,48	0,08	0,13	g5.1
Jacaranda copaia (BIGN)	304	40	20,14	26,74	0,37	0,17	0,17	g5.2
Inga alba (MIMO)	620	40	20,80	24,99	0,39	0,12	0,14	g5.2
Hevea guianensis (EUPH)	759	40	23,30	31,83	0,43	-0,01	0,08	g4.1
Inga cayennensis (MIMO)	802	40	16,73	18,78	0,59	0,28	0,14	g5.3
Eugenia cf. coffeifolia (MYRT)	636	39	15,02	17,19	NS	-0,02	0,06	g1.4
Parinari campestris (CHRY)	706	38	22,56	30,24	0,43	0,19	0,09	g5.2
Sclerobium melinonii (CAES)	210	37	32,18	44,40	0,27**	0,59	0,16	g3.2
Dendrobangia boliviana (ICAC)	622	37	20,31	27,06	0,42	0,06	0,07	g4.1
Vouacapoua americana (CAES)	116	36	32,43	42,46	0,24*	-0,16	0,20	g2.3
Swartzia guianensis (CAES)	609	36	13,82	15,76	NS	-0,22	0,07	g1.1

Ocotea sp1 (LAUR)	611	36	17,87	20,37	0,34**	0,08	0,12	g5.2
Rhedia acuminata (CLUS)	634	34	15,17	17,83	0,37	0,10	0,05	g5.2
Apeiba echinata et sp (TILI)	689	33	23,89	32,31	0,32**	0,19	0,15	g4.1
Talisia spp (SAPI)	624	31	14,00	15,12	0,33**	0,08	0,08	g5.2
Poecilanthe hostmanii (FABA)	637	29	19,80	23,71	NS	0,05	0,17	g1.4
Manilkara bidentata (SAPO)	205	28	25,77	33,10	0,38**	0,14	0,06	g4.1
Aspidosperma album (APOC)	217	28	22,28	30,88	0,30*	-0,05	0,12	g2.3
Mabea piri (EUPH)	651	27	18,41	18,62	0,60	0,15	0,08	g5.2
Miconia spp (MELA)	751	27	14,74	16,39	0,62	0,25	0,09	g5.3
Andira coriacea (FABA)	115	26	24,49	31,67	0,52	0,22	0,10	g4.2
Unonopsis guatterioides/rufescens	645	26	13,22	14,16	NS	-0,14	0,15	g1.2
Sacoglottis guianensis (HUMI)	648	26	25,80	33,74	NS	0,21	0,14	g2.4
Cordia cf. laevifrons (BORA)	772	26	14,35	17,03	0,42**	0,00	0,06	g5.1
Ecclinusa guianensis (SAPO)	613	25	21,49	27,53	NS	0,03	0,11	g2.3
Protium cf. heptaphyllum (BURS)	642	25	17,46	21,33	0,32*	0,21	0,16	g5.3
Caryocar glabrum (CARY)	209	24	32,71	45,68	NS	-0,07	0,21	g2.3
Hydrochorea corymbosa (MIMO)	632	24	25,01	32,15	0,34*	0,25	0,25	g3.1
Duroia aquatica (RUBI)	693	24	17,36	22,41	0,25*	0,01	0,07	g5.1
Inga huberi/stipularis (MIMO)	801	24	23,00	31,67	0,42**	0,15	0,11	g4.1
Simaba cf. multiflora (SIMA)	303	23	23,05	29,13	0,41**	0,03	0,14	g4.1
Swartzia panacoco (CAES)	686	23	21,99	28,17	NS	0,05	0,08	g2.3
Xylopia sp2 (ANNO)	688	23	20,72	24,43	0,39**	0,39	0,08	g5.3
Tapirira guianensis (ANAC)	656	22	20,65	25,78	0,53	0,36	0,25	g4.2
Guarea cf. guidonia/pubescens (MELI)	615	21	14,56	15,60	NS	-0,16	0,04	g1.1
Anacardium spruceanum (ANAC)	625	19	18,12	22,12	0,57**	0,17	0,10	g5.2
Ocotea cf. guianensis (LAUR)	646	19	16,55	20,69	NS	0,03	0,14	g5.1
Parkia velutina et spp (MIMO)	305	18	30,49	47,19	0,42*	0,41	0,26	g3.2
Enterolobium schomburgkii (MIMO)	101	7	22,84	29,92	-	0,19	0,09	g2.4
Peltogyne venosa (CAES)	102	7	24,41	33,26	-	0,05	0,21	g2.4
Bagassa guianensis (MORA)	104	1	52,12	72,89	-	0,31	-	g3.1
Lecythis zabucajo (LECY)	107	14	23,84	29,60	NS	-0,25	0,09	g2.1
Diploptropis purpurea (CAES)	108	10	28,24	38,83	NS	0,05	0,14	g2.4
Tabebuia serratifolia (BIGN)	110	6	22,60	32,31	-	0,24	0,28	g4.2
Vataireopsis sp (FABA)	111	3	15,62	20,29	-	0,01	0,31	g5.1

Pterocarpus rohrii (FABA)	112	12	25,80	27,50	NS	0,20	0,10	g4.2
Platonia insignis (CLUS)	113	12	31,52	43,13	NS	0,01	0,08	g2.4
Brosimum guianense (MORA)	114	14	22,70	26,74	NS	-0,25	0,02	g2.1
Dimorphandra polyandra (CAES)	202	1	38,73	55,39	-	0,15	.	g3.1
Terminalia amazonia (COMB)	203	4	27,65	35,33	-	0,03	0,15	g2.4
Balizia pedicellaris (MIMO)	204	9	42,45	54,59	-	0,19	0,21	g3.1
Cèdres divers (indéterminés) (LAUR) ^(a)	208	10	16,48	18,14	NS	-0,11	0,05	g1.2
Qualea rosea et Ruizteriana albiflora (VOCH) ^(a)	211	5	34,96	48,70	-	0,16	0,31	g3.1
Thyrsodium guianense (ANAC)	213	39	19,68	26,90	0,32**	0,10	0,17	g5.2
Parkia pendula (MIMO)	216	3	32,18	35,97	-	0,45	0,38	g4.2
Symphonia glob., Moronobea cocc. (CLUS) ^(a)	218	42	24,81	32,95	NS	0,01	0,15	g4.1
Hymenobium flavum (FABA)	219	2	23,03	30,24	-	-0,19	0,05	g2.2
Tetragastris sp (BURS) ^(a)	220	15	22,82	28,65	0,43*	-0,02	0,06	g4.1
Dipteryx odorata (FABA)	221	4	21,03	24,35	-	-0,04	0,11	g2.3
Glycidendron amazonicum (EUPH)	223	8	21,68	22,12	-	0,19	0,07	g5.3
Schefflera morototoni (ARAL)	302	7	24,78	30,56	-	0,04	0,13	g2.4
Brosimum rubescens (MORA)	306	14	22,20	31,19	0,48*	-0,08	0,12	g2.3
Laetia procera (FLAC)	309	21	24,08	30,88	NS	0,28	0,08	g4.2
Simarouba amara (SIMA)	315	2	31,04	41,54	-	0,70	0,03	g3.2
Virola surinamensis (MYRI)	316	3	25,89	32,95	-	0,27	0,41	g4.2
Divers indéterminés ^(b)	405	135	20,16	21,96	NS	-0,22	0,17	g1.2
Cordia cf. nodosa (BORA)	603	8	15,22	17,83	-	-0,26	0,04	g1.1
Ruizteriana albiflora (VOCH)	605	16	34,59	52,68	0,63**	0,22	0,06	g3.1
Couma guianensis (APOC)	612	12	23,69	30,24	0,54*	0,10	0,19	g4.1
Helicostylis pedunculata (MORA)	621	19	16,43	19,74	0,43*	-0,07	0,05	g5.1
Amanoa guianensis (EUPH)	628	12	19,24	25,07	NS	-0,13	0,08	g1.3
Antonia ovata (LOGA)	635	3	37,19	47,11	-	0,23	0,32	g3.1
Aspidosperma nitidum (APOC)	640	1	24,82	32,60	-	-0,24	.	g2.1
Aniba sp (LAUR)	647	7	13,19	13,05	-	-0,08	0,13	g1.4
Rinorea pectino-squamata (VIOL)	650	2	11,95	12,54	-	0,12	0,00	g5.2
Pourouma guianensis (MORA)	657	6	17,02	22,41	-	0,02	0,22	g5.2
Caraipa densifolia (CLUS)	660	5	19,09	24,51	-	-0,13	0,02	g1.3
Xylopia sp (ANNO)	661	10	17,64	19,66	NS	0,17	0,17	g5.2
Diospyros cf guianensis (EBEN)	663	4	12,71	12,73	-	0,00	0,03	g5.1

Hyeronima alchorneoides (EUPH)	664	12	24,72	29,60	NS	0,08	0,16	g2.4
Vismia cayennensis (CLUS)	665	7	17,02	20,37	-	0,29	0,21	g5.3
Pradosia ptychandra (SAPO)	668	3	15,53	17,63	-	0,01	0,05	g5.2
Luehopsis rugosa (TILI)	669	9	27,54	32,15	-	0,05	0,07	g4.1
Licaria cf rigida (LAUR)	670	1	24,19	23,87	-	-0,12	.	g1.3
Tetragastris cf. panamensis (BURS)	671	10	21,85	22,60	0,60*	0,02	0,09	g5.1
"Kissiki banda" (MYRT) ^(e)	672	1	14,48	17,83	-	-0,07	.	g1.4
Himatanthus articulatus (APOC)	673	4	17,75	23,56	-	-0,05	0,18	g1.4
Pterocarpus officinalis (FABA)	674	1	19,96	23,40	-	0,77	.	g5.3
Macoubea guianensis (APOC)	678	12	21,29	28,33	0,68**	0,04	0,06	g4.1
Xylopia cf nitida (ANNO)	680	12	18,88	25,94	0,61*	0,01	0,06	g5.2
Ocotea cf canaliculata (LAUR)	681	7	17,97	18,62	-	0,12	0,23	g5.3
Cecropia obtusa (MORA)	682	3	18,76	22,76	-	0,55	0,24	g5.3
"Gueli apici" (LAUR) ^(e)	684	15	15,01	16,55	NS	-0,19	0,15	g1.1
Byrsonima sp (MALP)	685	2	29,29	38,20	-	0,18	0,09	g3.1
Casearia cf decandra (FLAC)	691	1	10,98	10,98	-	-0,01	.	g5.1
Minquartia guianensis (OLAC)	694	4	31,86	37,69	-	0,10	0,11	g4.1
Heisteria densifrons (OLAC)	695	5	21,38	28,09	-	-0,13	0,09	g2.3
Tachigali paniculata (FABA)	696	5	32,68	45,82	-	1,09	0,10	g3.2
Palicourea guianensis (RUBI)	697	2	12,50	12,54	-	0,17	0,07	g5.3
"Weti bakouma" (SAPO) ^(e)	699	1	22,84	25,31	-	-0,17	.	g1.1
Licania cf canescens (CHRYS)	710	18	20,70	25,15	NS	-0,17	0,07	g1.2
"Sabana koko" (CHRYS) ^(e)	711	4	27,38	37,69	-	-0,04	0,04	g4.1
Couepia cf guianensis (CHRY)	712	6	31,95	38,83	-	-0,19	0,15	g2.3
Couepia cf bracteosa (CHRY)	718	2	31,41	34,12	-	0,39	0,02	g4.2
Vatairea cf guianensis (FABA)	720	2	26,29	33,90	-	0,56	0,51	g3.2
"Podotiki" ^(e)	724	5	18,22	22,44	-	0,21	0,15	g5.2
Inga sp. nov. (MIMO)	726	2	35,67	38,83	-	0,85	0,51	g3.2
Parahancornia fasciculata (APOC)	750	3	32,81	46,00	-	0,33	0,01	g3.2
Casearia arborea / javitensis (FLAC)	752	1	13,67	12,54	-	-0,23	.	g1.2
"Agui" ^(e)	753	4	23,39	32,60	-	0,09	0,17	g2.4
Licaria cf canella (LAUR)	754	8	18,29	21,17	-	-0,37	0,10	g1.1
Pisonia sp ou Neea sp (NYCT)	755	6	30,02	32,60	-	0,32	0,09	g4.2
Protium cf subserratum (BURS)	756	19	20,78	25,15	0,55**	0,03	0,10	g5.2

" <i>Papaoudou apici</i> " (LAUR) ^(a)	758	1	25,34	42,46	-	0,15		g2.4
Casearia cf. pitumba (FLAC)	761	11	15,58	17,35	NS	-0,04	0,07	g5.1
" <i>Pepe tiki</i> " (MYRT) ^(a)	762	10	15,19	17,57	NS	-0,30	0,03	g1.1
Luehopsis rosea (TILI)	764	2	21,71	25,31	-	0,19	0,31	g4.2
Pourouma sp (MORA)	765	2	16,98	19,89	-	0,02	0,27	g5.2
Vochysia cf. tomentosa (VOCHY)	766	1	46,08	73,93	-	1,40	.	g3.2
Brosimum cf. utile (MORA)	770	12	21,52	28,01	NS	-0,03	0,08	g2.3
Inga tubaeformis (MIMO)	803	6	16,63	17,67	-	0,05	0,16	g5.2
Inga sp. (MIMO)	804	9	20,78	25,31	-	0,38	0,20	g4.1
Inga sp. (MIMO)	805	1	19,15	16,23	-	1,10	.	g5.3
" <i>Baoulift</i> " ^(a)	815	3	17,97	22,41	-	0,13	0,29	g5.2

^(a) Codes attribués à l'origine à des arbres qui sont morts avant d'avoir pu être redéterminés (à partir de 1992).

^(b) Code d'origine des espèces "diverses". Toujours attribué à des arbres morts avant d'avoir pu être redéterminés, ou bien à des arbres non identifiés botaniquement.

^(c) Codes présents dans la base, attribués à des espèces non déterminées par Favrichon, ou bien absents de sa liste d'espèces (1995).

Tableau a9 (2 pages). Correspondance entre nos groupes et ceux de Favrichon (1994, 1995). 87 espèces ou groupes d'espèces comprenant plus de 18 individus dans le fichier d'ajustement ont contribué à l'élaboration de la classification : ils sont représentés en caractères gras. Les 86 espèces ou groupes d'espèces restant ont été rattachés a posteriori (10745 individus au total sont représentés). Enfin, 14 espèces ou groupes d'espèces supplémentaires présents dans la base de données de Paracou mais éliminés du fichier de base pour des raisons diverses (absence dans les huit parcelles de calibration, présence uniquement en bordures de parcelles, ou présentant diverses anomalies de mesures) ont également été rattachés, sur la base de la structure de leur population et de leur croissance moyenne annuelle entre 1988 et 1991 : ils figurent en italique dans le tableau. **NB** : tous les codes entre parenthèses concernent des espèces ou groupes d'espèces représentés par six individus et moins. Identification des groupes : voir texte principal.

Nouveaux groupes	Groupes de Favrichon (1994, 1995)				
	G1	G2	G3	G4	G5
g 1.1	404, 609, 615, 618, 641, 644 603, (684), 762 (666)	(699), 754			
g 1.2	222, 602, 614, 645, 679 710, (752)				
g 1.3	(660)	401 628, (670) 715			
g 1.4	313, 616, 623, 627, 636, 637, 687, 703 (672)	647, (673)			
g 2.1	105	114, (640)	107 (106)		
g 2.2	633	403, 659	(219)		
g 2.3	(695)	116, 217, 308, 402, 613, 617, 643, 686, 701, 705 306 (714)	209 (221), (712), 770 (225)	(690)	
g 2.4		619, 709 (758)	212, 224, 608, 648 101, 102, 108, 113	(203), 302, 664	
g 3.1			103, 214, 604, 698 605, (202)	632 (104), 204, (635), (685)	
g 3.2			(720), (750) 631, (768)	210, 305 (315), (696), (766) (301), (311)	(726)

Tableau a9 (fin).

g4.1		205, 206, 303, 622, 759 220, 612, 678, (694)	117	689, 801 669, 804	
g4.2	(718)	(755)	115, 607 (110)	207, 310, 656, 658 112, (216), 309, (316), (764) (215), 638	
g5.1	601, 606, 655, 693, 772 621, (663), (691), 761	312, 653, 662, 707 671	(111)	646	
g5.2	611, 624, 634, 651 (650) (719)	625, 629, 704 213, (668), (724), 756	706	304, 620 661, 680, (765)	(657), (803)
g5.3	(681)	223		688 (674), (805)	314, 642, 751, 802 665, (682), (697) 683

Ne figurent pas dans le tableau ci-dessus :

- « Agui » (code 753) : individus non classés par Favrichon, à classer en **g2.4**.
- « Baoulifi » (code 815) : individus non classés par Favrichon, à classer en **g5.2**.
- Cèdres divers (*Licaria spp.* + *Ocotea spp.* : code 208) : individus à classer en **g1.2** ; 208 est un ancien code qui n'est plus utilisé. Les individus concernés sont morts après 1991 et n'ont pu être redéterminés.
- Gaulettes (*Licania sp.* : code 711) : individus non classés par Favrichon, à classer en **g4.1**.
- Gonfolos rose et gris (*Qualea rosea* + *Ruizteriana albiflora* : code 211) : individus à classer en **g3.1**. 211 est un ancien code qui n'est plus utilisé (deux espèces distinguées). Individus morts après 1991, qui n'ont pu être redéterminés.
- Manils (*Symphonia globulifera* et *Moronobea coccinea* : code 218) : individus à classer en **g4.1**. 218 est un ancien code qui n'est plus utilisé (deux espèces distinguées).
- les 5 codes (501 à 505) concernant les Palmiers.

Enfin, divers indéterminés (code 405) ou individus dont l'identification est douteuse sont rattachés au groupe **g1.2**. Cela concerne 135 arbres présents dans le jeu de données de calibration. Huit codes douteux présents dans la base (hors fichiers d'ajustement et de validation) sont par ailleurs rattachés à ce groupe « divers ».

Tableau a10. Caractéristiques des écarts entre accroissements moyens annuels et prédictions par les différents modèles caractéristiques de nos 15 groupes de croissance, lorsque sont rattachés a posteriori à ces 15 groupes les 86 taxons présents dans le jeu de données d'ajustement et n'ayant pas participé à la classification. Les modèles de croissance utilisés sont ceux qui figurent dans le Tableau 4.7 du texte principal (s'y reporter également pour la signification des différents groupes). 10745 individus sont concernés.

Sous-groupe	Groupe g1.1	Groupe g1.2	Groupe g1.3	Groupe g1.4
Effectif	1183	991	560	922
Composition (codes représentés)	Base + 603, 684, 699, 754, 762	Base + 710, 752	Base + 628, 660, 670	Base + 647, 672, 673
Biais moyen (variance) (resp. Log(cm) et Log(cm) ²)	0,0007 (0,0640)	-0,0031 (0,0888)	-0,0010 (0,0836)	0,0005 (0,0588)
Normalité des erreurs (W) ^a	0,985 (NS)	0,973 ***	0,986 (NS)	0,988 (NS)
SCE ^b	75,70	87,88	46,73	54,15
SCT ^c	82,05	98,81	59,34	71,72
Pseudo-R ² ^d	0,077	0,111	0,213	0,245
Résultats d'une analyse de variance à un facteur « espèce » sur les résidus de chaque modèle				
F ^e	1,05 (NS)	0,55 (NS)	0,49 (NS)	1,13 (NS)

(suite)

Sous-groupe	Groupe g2.1	Groupe g2.2	Groupe g2.3	Groupe g2.4
Effectif	274	1253	1294	829
Composition (codes représentés)	Base + 107, 114, 640	Base + 219	Base + 221, 306, 695, 712, 770	Base + 101, 102, 108, 113, 203, 302, 664, 758
Biais moyen (variance) (resp. Log(cm) et Log(cm) ²)	0,0015 (0,0621)	-0,0004 (0,1092)	0,0008 (0,0928)	0,0026 (0,1184)
Normalité des erreurs (W) ^a	0,986 (NS)	0,989 (NS)	0,993 (NS)	0,992 (NS)
SCE ^b	16,94	136,70	120,00	98,03
SCT ^c	19,46	164,02	152,41	121,91
Pseudo-R ² ^d	0,129	0,167	0,213	0,196
Résultats d'une analyse de variance à un facteur « espèce » sur les résidus de chaque modèle				
F ^e	0,21 (NS)	0,13 (NS)	1,26 (NS)	0,88 (NS)

(suite)

Sous-groupe	Groupe g3.1	Groupe g3.2	Groupe g4.1	Groupe g4.2
Effectif	524	70	454	630
Composition (codes représentés)	Base + 104, 202, 204, 605, 635, 685	Base + 315, 696, 720, 726, 750, 766	Base + 113, 220, 612, 669, 678, 694, 804	Base + 110, 112, 216, 309, 316, 718, 755, 764
Biais moyen (variance) (resp. Log(cm) et Log(cm) ²)	0,0030 (0,1170)	0,0546 (0,1971)	0,0148 (0,0917)	0,0046 (0,1276)
Normalité des erreurs (W) ^a	0,962 ***	0,979 (NS)	0,988 (NS)	0,984 (NS)
SCE ^b	61,17	13,81	41,63	80,25
SCT ^c	89,46	19,98	71,95	124,23
Pseudo-R ² ^d	0,316	0,309	0,421	0,354
Résultats d'une analyse de variance à un facteur « espèce » sur les résidus de chaque modèle				
F ^e	0,79 (NS)	1,91 (NS)	1,08 (NS)	(NS)

(suite)

Sous-groupe	Groupe g5.1	Groupe g5.2	Groupe g5.3	Synthèse
Effectif	644	815	302	10745
Composition (codes représentés)	Base + 111, 621, 663, 671, 691, 761	Base + 213, 650, 657, 661, 668, 680, 724, 756, 765, 803	Base + 223, 665, 674, 681, 682, 697, 805	173 EGE
Biais moyen (variance) (resp. Log(cm) et Log(cm) ²)	0,0005 (0,0762)	-0,0014 (0,0998)	0,0057 (0,1067)	0,002 (0,093)
Normalité des erreurs (W) ^a	0,986 (NS)	0,989 (NS)	0,988 (NS)	0,022 (**)
SCE ^b	49,00	81,21	32,12	995,32
SCT ^c	79,93	130,61	58,86	1715,38
Pseudo-R ² ^d	0,387	0,378	0,454	0,420
Résultats d'une analyse de variance à un facteur « espèce » sur les résidus de chaque modèle				
F ^e	0,51 (NS)	1,03 (NS)	1,04 (NS)	1,09 (NS)

^a Test de Shapiro-Wilk (SAS Institute Inc., 1989). L'hypothèse de normalité ne peut être rejetée (NS), est rejetée avec un risque d'erreur de 5% (*), de 1% (**), de 1‰ (***)

^b Somme des carrés des erreurs.

^c $\sum (y_i - \bar{y})^2$.

^d $1 - (SCR / \sum (y_i - \bar{y})^2)$.

^e *** : $p \leq 0,001$.

Tableau a11. Analyse de variance hiérarchique pondérée « Parcelle (Traitement) » des résidus du modèle à 65 paramètres prenant en compte les 15 groupes d'espèces. Analyse réalisée avec la PROC GLM de SAS (Statistical Analysis System, 1989), qui prend en compte les plans déséquilibrés (10745 arbres étudiés).

Source	ddl	SCE	CME	F
Modèle	7	0,0139	0,0020	28,82 (***)
Traitement	3	0,0060	0,0020	1,00 (NS) ^a
Parcelle (Traitement)	4	0,0079	0,0020	
Erreur	10737	0,7406	0,0001	
Total corrigé	10744	0,7545		

* : $0,01 < p(>F) \leq 0,05$. ** : $0,001 < p(>F) \leq 0,01$. *** : $p(>F) \leq 0,001$.

^a $F = CME_{\text{Traitement}} / CME_{\text{Parcelle(Traitement)}}$, comparé à $F_{0,05}(3,4) = 6,59$.

Tableau a12. Evolution, sur les huit parcelles d'ajustement, des deux indices ΔNBT et ΔSTT au cours de deux périodes successives de trois ans. Dans chaque cellule figurent : (i) la valeur moyenne de l'indice, (ii) son (*écart-type*) et (iii) le coefficient de corrélation linéaire de Pearson entre le logarithme de l'accroissement annuel lissé entre 1988 et 1991 et l'indice étudié.

N° de la parcelle (traitement)	Effectif	ΔNBT		ΔSTT	
		1985-1988	1988-1991	1985-1988	1988-1991
1 (T0)	1596	-4,9 (3,6) <i>NS</i>	1,2 (4,1) 0,11 (***)	-0,24 (0,50) <i>NS</i>	0,14 (0,36) <i>NS</i>
6 (T0)	1561	-1,9 (3,3) <i>NS</i>	1,2 (4,5) 0,07 (**)	0,01 (0,22) <i>NS</i>	0,08 (0,39) <i>NS</i>
7 (T1)	1376	-28,2 (10,8) <i>NS</i>	1,9 (5,2) 0,06 (*)	-1,41 (0,79) -0,09 (**)	0,13 (0,35) <i>NS</i>
9 (T1)	1344	-28,8 (14,3) -0,17 (***)	2,1 (5,3) 0,19 (***)	-1,43 (0,79) -0,25 (***)	0,14 (0,21) 0,07 (*)
5 (T2)	1449	-28,7 (13,3) -0,15 (***)	-6,5 (8,3) 0,07 (*)	-2,44 (0,97) -0,15 (***)	-0,48 (0,58) <i>NS</i>
10 (T2)	1177	-34,2 (10,4) <i>NS</i>	3,3 (7,5) 0,09 (**)	-3,24 (1,05) <i>NS</i>	-0,28 (0,59) <i>NS</i>
4 (T3)	992	-59,0 (17,5) <i>NS</i>	4,6 (6,8) 0,08 (**)	-3,80 (1,06) -0,07 (*)	-0,14 (0,39) <i>NS</i>
12 (T3)	1250	-40,7 (14,6) -0,16 (***)	2,0 (6,8) 0,22 (***)	-3,22 (1,31) -0,18 (***)	-0,25 (0,60) 0,18 (***)

* : coefficient de corrélation linéaire significativement différent de 0 avec $0,01 \leq p < 0,01$,
 ** $0,001 \leq p < 0,0001$ et *** $p \geq 0,0001$. *NS* : coefficient non significativement différent de 0.

Tableau a13. Valeurs de la moyenne et de la variance des deux variables « accroissement moyen annuel lissé » entre 1988 et 1991 et « résidu du modèle de croissance » (modèle complet à 65 paramètres) en fonction des différentes modalités utilisées dans l'ACM « pédologie » (voir § 4.5.3.5). Seuls les individus des parcelles 1 et 6 (3157) sont utilisés.

Variable	Effectif	Accroissement moyen annuel		Résidu du modèle complet à 65 paramètres	
		Moyenne (cm)	Variance (cm ²)	Moyenne (Log(cm))	Variance (Log(cm) ²)
PARCELLE					
<i>P1</i>	1596	0,14	0,03	0,021	0,078
<i>P6</i>	1561	0,13	0,03	0,013	0,078
<i>P11 (pour mémoire)</i>	2025	0,11	0,02	0,007	0,076
DRAINAGE					
<i>drain1</i>	781	0,16	0,05	0,046	0,087
<i>drain2</i>	958	0,13	0,03	0,025	0,078
<i>drain3</i>	817	0,12	0,03	-0,015	0,074
<i>drain4</i>	576	0,12	0,02	0,007	0,071
<i>drain5</i>	25	0,13	0,01	0,017	0,061
HYDROM.					
<i>hydro0</i>	1475	0,13	0,03	0,006	0,076
<i>hydro1</i>	1682	0,14	0,03	0,027	0,080
SABLE					
<i>sable0</i>	1928	0,12	0,02	0,005	0,073
<i>sable1</i>	1229	0,16	0,04	0,036	0,085
PH					
<i>pH1</i>	747	0,15	0,04	0,042	0,084
<i>pH2</i>	1671	0,13	0,03	0,010	0,078
<i>pH3</i>	739	0,12	0,03	0,007	0,071
CARBONE					
<i>carbo1</i>	382	0,17	0,05	0,067	0,082
<i>carbo2</i>	2337	0,13	0,03	0,007	0,076
<i>carbo3</i>	438	0,14	0,03	0,027	0,083
NODULE					
<i>nod0</i>	2238	0,14	0,03	0,028	0,080
<i>nod1</i>	919	0,12	0,02	-0,009	0,072

Tableau a14. Valeurs de la moyenne et de la variance de la variable « résidu du modèle de croissance » (modèle complet à 65 paramètres, prenant en compte les groupes d'espèces) en fonction des parcelles et des différentes modalités de la variable « topographie ». NB : les parcelles indépendantes (indiquées entre parenthèses) sont reportées à titre indicatif, pour comparaison.

Traitement	Parcelle	Topographie	Effectif concerné	Résidus du modèle complet prenant les groupes d'espèces en compte	
				Moyenne (Log(cm))	Variance (Log(cm) ²)
T0	P1	Bas-fond	190	0,075	0,098
		Versant	1320	0,010	0,076
		Sommet	86	0,080	0,058
	P6	Bas-fond	250	0,040	0,085
		Versant	1207	0,012	0,075
		Sommet	104	-0,043	0,096
	(P11)	Bas-fond	27	0,117	0,168
		Versant	1692	-0,005	0,075
		Sommet	306	0,062	0,067
T1	(P2)	Bas-fond	147	0,018	0,115
		Versant	1227	0,044	0,102
		Sommet	90	0,054	0,107
	P7	Bas-fond	138	-0,006	0,119
		Versant	1067	-0,018	0,079
		Sommet	171	-0,100	0,078
	P9	Bas-fond	13	-0,028	0,165
		Versant	1210	-0,017	0,092
		Sommet	121	-0,025	0,082
T2	(P3)	Bas-fond	40	0,002	0,074
		Versant	913	0,094	0,112
		Sommet	34	0,050	0,126
	P5	Bas-fond	88	-0,010	0,112
		Versant	1306	-0,024	0,095
		Sommet	55	-0,026	0,126
	P10	Bas-fond	43	0,152	0,138
		Versant	1086	0,098	0,112
		Sommet	48	0,088	0,127
T3	P4	Bas-fond	28	0,041	0,124
		Versant	933	-0,040	0,111
		Sommet	31	-0,071	0,091
	(P8)	Bas-fond	5	0,078	0,066
		Versant	745	0,065	0,125
		Sommet	47	0,092	0,119
	P12	Bas-fond	1	0,034	-
		Versant	1096	-0,015	0,083
		Sommet	153	-0,030	0,127

Tableau a15. Analyse de variance à deux facteurs croisés (topographie, parcelle) avec interaction des résidus du modèle complet à 65 paramètres. Analyse réalisée avec la PROC GLM de SAS (Statistical Analysis System, 1989), qui prend en compte les plans déséquilibrés. 10745 arbres étudiés.

Source	ddl	SCE	CME	F
Modèle	23	19,695	0,856	9,41 (***)
« Type I SS »				
Topographie	2	1,784	0,892	9,80 (***)
Parcelle	7	16,264	2,323	25,53 (***)
Topo*Parcelle	14	1,647	0,118	1,29 (NS)
« Type III SS »				
Topographie	2	0,227	0,113	1,25 (NS)
Parcelle	7	5,339	0,763	8,38 (***)
Topo*Parcelle	14	1,648	0,118	1,29 (NS)
Erreur	10721	975,598	0,091	
Total corrigé	10744	995,293		

* : $0,01 < p(>F) \leq 0,05$. ** : $0,001 < p(>F) \leq 0,01$. *** : $p(>F) \leq 0,001$.

Tableau a16. Valeurs de la moyenne et de la variance de la variable « résidu du modèle de croissance » (modèle complet à 65 paramètres, prenant en compte les groupes d'espèces) en fonction des 15 groupes d'espèces définis au paragraphe 4.5.3.3 et des différentes modalités de la variable « topographie ». Statistiques portant sur les 10745 arbres du jeu de calibration.

Groupe d'espèces	Topographie	Effectif concerné	Résidus du modèle complet à 65 paramètres	
			Moyenne (Log(cm))	Variance (Log(cm) ²)
g1.1	Bas-fond	111	0,007	0,067
	Versant	1006	0,000	0,063
	Sommet	66	-0,001	0,072
g1.2	Bas-fond	51	-0,011	0,093
	Versant	874	0,000	0,090
	Sommet	66	-0,044	0,067
g1.3	Bas-fond	35	0,023	0,049
	Versant	492	0,005	0,081
	Sommet	33	-0,115	0,153
g1.4	Bas-fond	74	0,036	0,073
	Versant	800	-0,003	0,057
	Sommet	48	0,008	0,076
g2.1	Bas-fond	11	-0,031	0,039
	Versant	232	0,000	0,064
	Sommet	31	0,028	0,061
g2.2	Bas-fond	53	0,136	0,159
	Versant	1082	-0,001	0,105
	Sommet	118	-0,055	0,114
g2.3	Bas-fond	62	0,067	0,107
	Versant	1113	0,000	0,093
	Sommet	119	-0,024	0,078
g2.4	Bas-fond	71	0,055	0,132
	Versant	690	0,000	0,119
	Sommet	68	-0,027	0,093
g3.1	Bas-fond	30	-0,090	0,091
	Versant	465	0,011	0,117
	Sommet	29	-0,030	0,138
g3.2	Bas-fond	12	-0,010	0,104
	Versant	50	-0,024	0,213
	Sommet	8	-0,026	0,272
g4.1	Bas-fond	37	-0,009	0,098
	Versant	392	0,014	0,092
	Sommet	25	0,059	0,078
g4.2	Bas-fond	82	0,053	0,168
	Versant	514	0,005	0,116
	Sommet	34	-0,118	0,199
g5.1	Bas-fond	42	0,007	0,086
	Versant	562	0,003	0,075
	Sommet	40	-0,035	0,086
g5.2	Bas-fond	44	0,131	0,102
	Versant	711	-0,011	0,099
	Sommet	60	0,012	0,093
g5.3	Bas-fond	36	0,065	0,111
	Versant	242	0,005	0,103
	Sommet	24	-0,080	0,130

Tableau a17. Principaux résultats de l'ajustement, à sept de nos groupes d'espèces, de modèles spécifiques prenant en compte la variable « topographie ». Ces résultats doivent être comparés à ceux présentés dans le Tableau 4.7 du texte principal.

Sous-groupe	Groupe g1.3	Groupe g2.2	Groupe g2.3	Groupe g2.4
Effectif	542	1251	1253	765
<i>Facteur modificateur</i> ^a	(6)	(5)	(5)	(5)
Paramètres	Estimation (σ)	Estimation (σ)	Estimation (σ)	Estimation (σ)
a' ₁	-4,377 (0,079)	-4,379 (0,053)	-4,391 (0,047)	-4,444 (0,059)
a' ₂	-4,382 (0,059)	-4,525 (0,031)	-4,456 (0,027)	-4,510 (0,045)
a' ₃	-4,512 (0,079)	-4,579 (0,042)	-4,481 (0,040)	-4,537 (0,062)
K (cm)	93,74 (7,57)	94,70 (2,85)	97,04 (2,33)	112,56 (4,53)
b (*10 ⁻³)	0,00	0,00	0,00	0,00
c (*10 ⁻³)	-15,97 (1,37)	-22,59 (1,54)	-21,12 (1,62)	-31,35 (2,79)
d (*10 ⁻³)	-0,055 (0,014)	-0,176 (0,027)	-0,157 (0,030)	-0,289 (0,054)
SCR	44,95	135,22	115,73	89,31
SCT	57,76	163,94	147,36	112,50
$\hat{\sigma}^2$ (Log(cm) ²)	0,084	0,109	0,093	0,118
R ²	0,222	0,175	0,215	0,206

(suite)

Sous-groupe	Groupe g4.2	Groupe g5.2	Groupe g5.3
Effectif	575	708	273
<i>Facteur modificateur</i> ^a	(1)	(3)	(2)
Paramètres	Estimation (σ)	Estimation (σ)	Estimation (σ)
a' ₁	-3,986 (0,170)	-2,995 (0,170)	-3,756 (0,360)
a' ₂	-4,055 (0,164)	-3,147 (0,160)	-3,802 (0,364)
a' ₃	-4,213 (0,176)	-3,120 (0,167)	-3,854 (0,373)
K (cm)	98,83 (12,17)	74,90 (3,78)	96,66 (30,99)
b (*10 ⁻³)	-27,78 (10,36)	-224,67 (30,16)	-44,80 (19,10)
c (*10 ⁻³)	-31,06 (2,62)	-17,97 (1,16)	-21,85 (1,89)
d (*10 ⁻³)	-0,208 (0,042)	-0,055 (0,011)	-0,106 (0,021)
SCR	71,72	68,51	28,33
SCT	115,44	110,36	50,02
$\hat{\sigma}^2$ (Log(cm) ²)	0,126	0,098	0,106
R ²	0,379	0,379	0,434

^a Voir texte principal et Tableau 4.7.

Tableau a18. Caractéristiques des écarts entre valeurs observées et valeurs prédites, par classes de diamètre, pour chacun des deux jeux de données de calibration et indépendant.

(a) : Valeurs prédites par le modèle [13].

Classes de diamètre (cm)	Calibration			Indépendant		
	Effectif	Moyenne	Variance	Effectif	Moyenne	Variance
10-12	1850	-0,008	0,090	913	0,034	0,083
12-14	1582	-0,002	0,100	765	0,043	0,107
14-16	1349	0,021	0,117	683	0,047	0,110
16-18	1052	0,006	0,119	437	0,002	0,126
18-20	771	-0,020	0,118	372	0,038	0,127
20-22,5	800	0,019	0,131	374	0,068	0,134
22,5-25	636	0,034	0,144	304	0,049	0,146
25-30	920	0,003	0,133	440	0,069	0,147
30-35	657	0,013	0,149	291	0,015	0,165
35-40	517	-0,059	0,164	228	-0,011	0,163
40-45	250	-0,026	0,153	101	-0,032	0,144
45-50	150	-0,003	0,189	68	0,059	0,202
≥ 50	211	0,029	0,237	99	0,125	0,190

(b) : Valeurs prédites par le modèle complet à 65 paramètres faisant intervenir les groupes d'espèces.

Classes de diamètre (cm)	Calibration			Indépendant		
	Effectif	Moyenne	Variance	Effectif	Moyenne	Variance
10-12	1850	0,012	0,077	913	0,057	0,075
12-14	1582	0,004	0,075	765	0,051	0,087
14-16	1349	0,021	0,085	683	0,049	0,080
16-18	1052	0,006	0,082	437	0,021	0,087
18-20	771	-0,023	0,085	372	0,049	0,090
20-22,5	800	0,007	0,092	374	0,066	0,112
22,5-25	636	0,007	0,104	304	0,040	0,103
25-30	920	-0,010	0,100	440	0,047	0,114
30-35	657	-0,008	0,116	291	0,017	0,135
35-40	517	-0,051	0,120	228	-0,005	0,139
40-45	250	-0,020	0,117	101	-0,072	0,112
45-50	150	-0,009	0,143	68	0,024	0,191
≥ 50	211	0,042	0,212	98	0,009	0,177

Tableau a19. Estimation (et erreur-standard asymptotique) des six paramètres du modèle [13] sur 13 jeux de données de calibration différents : un jeu complet comprenant l'intégralité des données (12 parcelles) et 12 jeux constitués de 11 parcelles, une parcelle différente étant éliminée à chaque fois.

Groupes de parcelles	Effectif concerné	Paramètres : estimation (écart-type asymptotique)					
		a'	m	K	b (*10 ⁻³)	c (*10 ⁻³)	d (*10 ⁻³)
Toutes (12)	15820	-5,35 (0,10)	1,144 (0,150)	223 (27)	-93,90 (11,95)	-18,64 (0,48)	-0,124 (0,008)
- P1	14224	-5,32 (0,10)	1,272 (0,188)	205 (25)	-93,39 (12,66)	-18,60 (0,52)	-0,121 (0,008)
- P2	14554	-5,27 (0,10)	1,071 (0,132)	218 (25)	-103,98 (12,19)	-18,23 (0,49)	-0,124 (0,008)
- P3	14833	-5,39 (0,12)	1,034 (0,132)	242 (32)	-93,77 (12,22)	-18,26 (0,49)	-0,124 (0,008)
- P4	14828	-5,40 (0,10)	1,306 (0,187)	211 (25)	-80,93 (12,43)	-18,67 (0,53)	-0,101 (0,010)
- P5	14371	-5,23 (0,10)	1,108 (0,138)	213 (24)	-110,18 (12,58)	-18,50 (0,50)	-0,128 (0,008)
- P6	14259	-5,57 (0,11)	1,419 (0,266)	230 (37)	-64,72 (12,97)	-19,35 (0,54)	-0,128 (0,008)
- P7	14444	-5,38 (0,13)	1,004 (0,134)	246 (35)	-98,74 (12,42)	-18,78 (0,50)	-0,131 (0,008)
- P8	15023	-5,30 (0,11)	1,074 (0,133)	224 (27)	-104,20 (12,43)	-18,68 (0,50)	-0,126 (0,009)
- P9	14476	-5,40 (0,11)	1,115 (0,156)	235 (32)	-88,21 (12,42)	-18,58 (0,50)	-0,124 (0,008)
- P10	14643	-5,40 (0,11)	1,067 (0,139)	238 (31)	-89,90 (12,24)	-17,89 (0,49)	-0,118 (0,008)
- P11	13795	-5,35 (0,10)	1,460 (0,238)	194 (22)	-90,92 (12,77)	-19,30 (0,55)	-0,128 (0,009)
- P12	14570	-5,34 (0,11)	1,072 (0,139)	231 (29)	-98,56 (12,73)	-18,95 (0,50)	-0,131 (0,008)

Tableau a20. Calcul de quelques statistiques d'évaluation sur les différents jeux de données utilisés dans le cadre du « *jackknife* ». Les paramètres du modèle [13] ont été estimés sur 13 jeux de données différents, ils sont présentés dans le Tableau a19. Dans 12 cas, il est possible de réaliser une évaluation quantitative sur la parcelle éliminée du jeu de calibration. La colonne « type de données » précise si le jeu a servi à la calibration (« C ») ou non (« I »). NB. Les notations utilisées dans les colonnes « statistiques » sont reprises du paragraphe 4.6.1.

Parcelles	Type de données	Effectif	Statistiques d'évaluation				
			B	T	MSE	Pseudo-R ²	J ²
Toutes	C	15820	0,000	0,00	0,123	0,257	-
Toutes - P1	C	14224	0,000	0,00	0,127	0,250	
P1	I	1596	0,042	5,41 (***)	0,096	0,227	0,756
Toutes - P2	C	14554	0,000	0,00	0,122	0,260	
P2	I	1266	0,044	4,18 (***)	0,143	0,217	1,175
Toutes - P3	C	14833	0,000	0,00	0,121	0,253	
P3	I	987	0,048	3,82 (***)	0,158	0,176	1,301
Toutes - P4	C	14828	0,000	0,00	0,120	0,270	
P4	I	992	-0,135	-10,76 (***)	0,175	-0,051	1,454
Toutes - P5	C	14371	0,000	0,00	0,122	0,265	
P5	I	1449	-0,031	-3,21 (**)	0,134	0,169	1,096
Toutes - P6	C	14259	0,000	0,00	0,126	0,251	
P6	I	1561	0,029	3,74 (***)	0,097	0,136	0,770
Toutes - P7	C	14444	0,000	0,00	0,125	0,264	
P7	I	1376	-0,047	-5,26 (***)	0,111	0,156	0,887
Toutes - P8	C	15023	0,000	0,00	0,122	0,254	
P8	I	797	-0,010	-0,69 (NS)	0,157	0,052	1,293
Toutes - P9	C	14476	0,000	0,00	0,123	0,264	
P9	I	1344	-0,036	-3,80 (***)	0,124	0,175	1,001
Toutes - P10	C	14643	0,000	0,00	0,120	0,254	
P10	I	1177	0,078	6,62 (***)	0,168	0,142	1,400
Toutes - P11	C	13795	0,000	0,00	0,129	0,239	
P11	I	2025	0,033	5,10 (***)	0,087	0,137	1,491
Toutes - P12	C	14570	0,000	0,00	0,123	0,261	
P12	I	1250	-0,049	-4,99 (***)	0,125	0,178	1,014

Tableau a21. Importance des différents types de mortalité observés sur la période 1988-1991 sur les parcelles de calibration. Deux cas ont été considérés : 1) l'échantillon de peuplement sur lequel nous ajustons les modèles : il s'agit d'individus situés à 30 m des bords des parcelles, vivants entre 1985 et 1988 et ne présentant pas d'anomalies dans leurs mesures et 2) le peuplement total présent sur les parcelles en début de période d'étude (1988) et dont est issu l'échantillon. Dans chaque cellule figure en gras le pourcentage de morts (ou de vivants à l'issue de la période 1988-1991) par rapport au nombre total d'individus présents et vivants en début de période d'étude (1988) ; on a fait figurer entre parenthèses l'effectif concerné ramené à l'hectare, et à l'année en ce qui concerne les individus morts entre 1988 et 1991. Voir texte principal pour davantage de détails.
MSP : morts sur pied. CHA1 : chablis primaires. CHA2 : chablis secondaires.

Traitement	Vivants en 1991 (ha)		MSP (ha/an)		CHA1 (ha/an)		CHA2 (ha/an)		Mortalité totale (ha/an)	
Parcelles de calibration										
	Jeu de données étudié	Total parcelles	Jeu de données étudié	Total parcelles	Jeu de données étudié	Total parcelles	Jeu de données étudié	Total parcelles	Jeu de données étudié	Total parcelles
T0 (P1, P6)	96,81 % (260,16)	97,08 % (569,44)	0,64 % (1,73)	0,57 % (3,33)	0,34 % (0,91)	0,29 % (1,73)	0,08 % (0,21)	0,11 % (0,64)	1,07 % (2,85)	0,97 % (5,70)
T1 (P7, P9)	96,93 % (224,72)	95,78 % (489,68)	0,46 % (1,07)	0,63 % (3,23)	0,36 % (0,83)	0,53 % (2,69)	0,21 % (0,48)	0,25 % (1,28)	1,03 % (2,38)	1,41 % (7,20)
T2 (P5, P10)	93,39 % (215,76)	91,47 % (475,44)	0,69 % (1,59)	1,04 % (5,41)	0,61 % (1,41)	0,91 % (4,75)	0,90 % (2,08)	0,89 % (4,61)	2,20 % (5,08)	2,84 % (14,77)
T3 (P4, P12)	95,03 % (188,08)	93,94 % (431,68)	0,59 % (1,17)	0,82 % (3,76)	0,40 % (0,80)	0,66 % (3,04)	0,66 % (1,31)	0,54 % (2,48)	1,65 % (3,28)	2,02 % (9,28)
Total	95,61 % (222,18)	94,66 % (491,56)	0,60 % (1,39)	0,76 % (3,93)	0,42 % (0,99)	0,59 % (3,05)	0,44 % (1,02)	0,43 % (2,25)	-	-

Tableau a22. Données du jeu de calibration constitué pour l'étude du recrutement. On a fait figurer, pour chaque traitement, le nombre de placeaux sur lesquels ont été respectivement recrutés 0, 1, 2, 3, 4 ou 5 individus au cours de la période 1988-1991. Huit parcelles et 529 placeaux de 10m*10m par parcelle sont concernés.

Traitement	Nombre d'individus recrutés par placeau						Total placeaux
	0	1	2	3	4	5	
Témoins	873	159	23	2	0	1	1058
T1	799	209	43	5	2	0	1058
T2	721	261	62	10	3	1	1058
T3	714	253	69	17	3	2	1058
Tous	3107	882	197	34	8	4	4232

ANNEXE 7

Description du simulateur SELVA

SOMMAIRE

1 - Paramétrage et lancement d'une simulation à l'aide de l'interface.....	1
<i>1.1 - Accès au simulateur.....</i>	<i>1</i>
<i>1.2 - Paramétrage.....</i>	<i>1</i>
1.2.1 - Définition des groupes d'espèces.....	1
1.2.2 - Affectation de couleurs de visualisation.....	1
1.2.3 - Caractéristiques de la simulation.....	4
1.2.4 - Choix du type et de la configuration des fichiers de sortie.....	4
<i>1.3 - Lancement d'une simulation.....</i>	<i>4</i>
2 - Description des classes de SELVA.....	5
<i>2.1 - Les fenêtres.....</i>	<i>5</i>
2.1.1 - ForestLauncher.....	5
2.1.2 - FenetreMain.....	5
2.1.3 - FenetreForet.....	9
2.1.4 - FenetreMainEvolution.....	10
2.1.5 - FenetreProcess.....	11
2.1.6 - FenetreSelectionListe.....	11
2.1.7 - FenetreEspèces.....	11
2.1.8 - FenetreOptionsCouleurs.....	12
2.1.9 - FenetreChoixCouleur.....	13
2.1.10 - FenetrePlaceau.....	13
2.1.11 - FenetreEclaircie.....	14
2.1.12 - FenetreDiametre.....	14
2.1.13 - FenetreEvolution.....	15
2.1.14 - FenetreArbre.....	15
2.1.15 - FenetreLegende.....	16
<i>2.2 - Les MVC (Model / View / Controller).....</i>	<i>16</i>
2.2.1 - ForestEnvironmentView.....	16
2.2.2 - ForestEnvironmentController.....	18
2.2.3 - GenericView.....	18
2.2.4 - PlaceauView.....	18
2.2.5 - PlaceauController.....	19
2.2.6 - ArrayView.....	19
2.2.7 - ArrayViewAvecMoyenne.....	20
2.2.8 - TreeView.....	20
2.2.9 - EffectifsDiametreView.....	21
2.2.10 - EvolutionView.....	21
<i>2.3 - Les modèles.....</i>	<i>21</i>
2.3.1 - EnvironmentForest.....	21
2.3.1.1 - <u>La méthode act</u>	22

2.3.1.2 - <u>Les sorties de fichiers</u>	23
2.3.1.3 - <u>Le point sur : la création et l'initialisation des placeaux</u>	25
2.3.1.4 - <u>Le point sur : la création et l'initialisation des arbres</u>	26
2.3.1.5 - <u>Les calculs toroïdaux</u>	26
2.3.1.6 - <u>Variables d'instance</u>	27
2.3.2 - Placeau	29
2.3.3 - Evolution	32
2.3.4 - ForestFile	34
2.3.5 - ForestKernelScheduler	35
2.4- Les arbres	36
2.4.1 - EspeceGrise	36
2.4.1.1 - <u>Le point sur : le destin des arbres</u>	37
2.4.1.2 - <u>Le point sur : le cas des éclaircies</u>	39
2.4.1.3 - <u>Le point sur : le cas des recrutés</u>	40
2.4.1.4 - <u>Autres méthodes et variables d'instance</u>	40
2.4.2 - TableauEvolutionArbre	42
2.4.3 - ModuleInstanceUnique	42
2.4.4 - DetectionGrise	42
2.4.5 - Deliberation	43
2.4.6 - DeliberationGrise	43
2.4.7 - DeliberationAngelique	43
2.4.8 - DeliberationPetitGris	44
2.4.9 - EffectationGrise	44
2.4.10 - EffectationAngelique	45
2.4.11 - EffectationPetitGris	46
2.5 - Les stades juvéniles	46
2.5.1 - StadeJuvenile	47
2.5.2 - StadeJuvenileAngelique	48
2.5.3 - Population	48
2.5.4 - PopulationEffectifsNumeriques	49
2.5.5 - PopulationEffectifsCollectifs	49
2.5.6 - PopulationEffectifsIndividuels	49
2.5.7 - PopulationGraines	49
2.5.8 - PopulationPlantules	49
2.5.9 - PopulationGaules	50
2.5.10 - PopulationPerches	50
2.6 - Les "outils" de SELVA	50
2.6.1 - Options	50
2.6.2 - Tools	52
2.6.3 - WriteFile	53
2.6.4 - FileAsArray	53
2.6.5 - ForestRandomUnif	54
2.6.6 - Gauss	54
2.6.7 - Gauss1	54
2.6.8 - ForestTime	54

1 - Paramétrage et lancement d'une simulation à l'aide de l'interface

Les différentes étapes sont présentées sur les Fig.1 et 2 et le texte ci-dessous reprend les numéros désignant les fenêtres.

1.1 - Accès au simulateur

L'accès aux fonctionnalités de VisualWorks 1.0 en général, et de SELVA en particulier, se fait par l'intermédiaire de la fenêtre « Selva Launcher » (1). Le menu de la fenêtre permet :

- « **Simul** » :
 - de « rejouer » de vieilles simulations précédemment sauvegardées ;
 - de réinitialiser deux dictionnaires importants de SELVA (voir § 2.6.1 ci-dessous) ;
 - de nettoyer la mémoire après une série de simulations ;
 - de sauvegarder et/ou de recharger le code de SELVA.
- « **Options** » :
 - d'avoir accès aux fenêtres de paramétrage « Regroupement d'espèces » et « Options couleurs » (3) et (4), sur lesquelles nous revenons plus loin ;
 - de sauvegarder différentes options dans les deux dictionnaires principaux de SELVA, ou de rappeler une sauvegarde précédemment effectuée.
- « **SmallTalk** » : d'accéder aux fonctionnalités de VisualWorks 1.0.

Le cliquage sur l'icône de lancement <Selva> provoque l'ouverture d'une deuxième fenêtre de lancement : (2) à partir de laquelle se fait l'essentiel du paramétrage initial.

1.2 - Paramétrage

1.2.1 - Définition des groupes d'espèces

La manière dont on veut utiliser l'information « espèce » contenue dans les fichiers des parcelles doit être définie au niveau de la fenêtre « Regroupement d'espèces » (3), à laquelle on accède par le menu « Options » de (2) (fenêtre « Lancement »).

Les libellés des EGE sont saisis par l'utilisateur grâce au bouton <Ajout> de la liste « Libelles groupe » et les codes espèce correspondant grâce au bouton <Ajout> de la liste « Codes essence ». Par défaut, le libellé « EspeceGrise » contient tous les codes non affectés à d'autres libellés.

Une fois les libellés saisis et les codes affectés, il faut associer à chaque EGE défini :

- un module de délibération spécifique, chargé notamment de l'évaluation des modèles de mortalité et de croissance qui lui sont propres ;
- un booléen provoquant ou non le calcul d'une autocorrélation entre valeurs successives de la croissance individuelle (voir § 7.2.2.3, chapitre 7 du texte principal) ;
- une couleur de visualisation, dont le choix peut être fait par l'intermédiaire de la fenêtre « Couleurs » : (5). Cette couleur est utilisée pour la représentation des individus sur le plan apparaissant dans la fenêtre (8) « Parcelle » (voir Fig.2).

1.2.2 - Affectation de couleurs de visualisation

Parallèlement au système précédent, on peut ré-affecter des couleurs aux EGE et d'une manière plus générale à tous les éléments composant le peuplement vivant ou mort d'une parcelle à l'aide de la fenêtre « Options couleurs » (4) à laquelle on accède par le menu « Options » de (2).

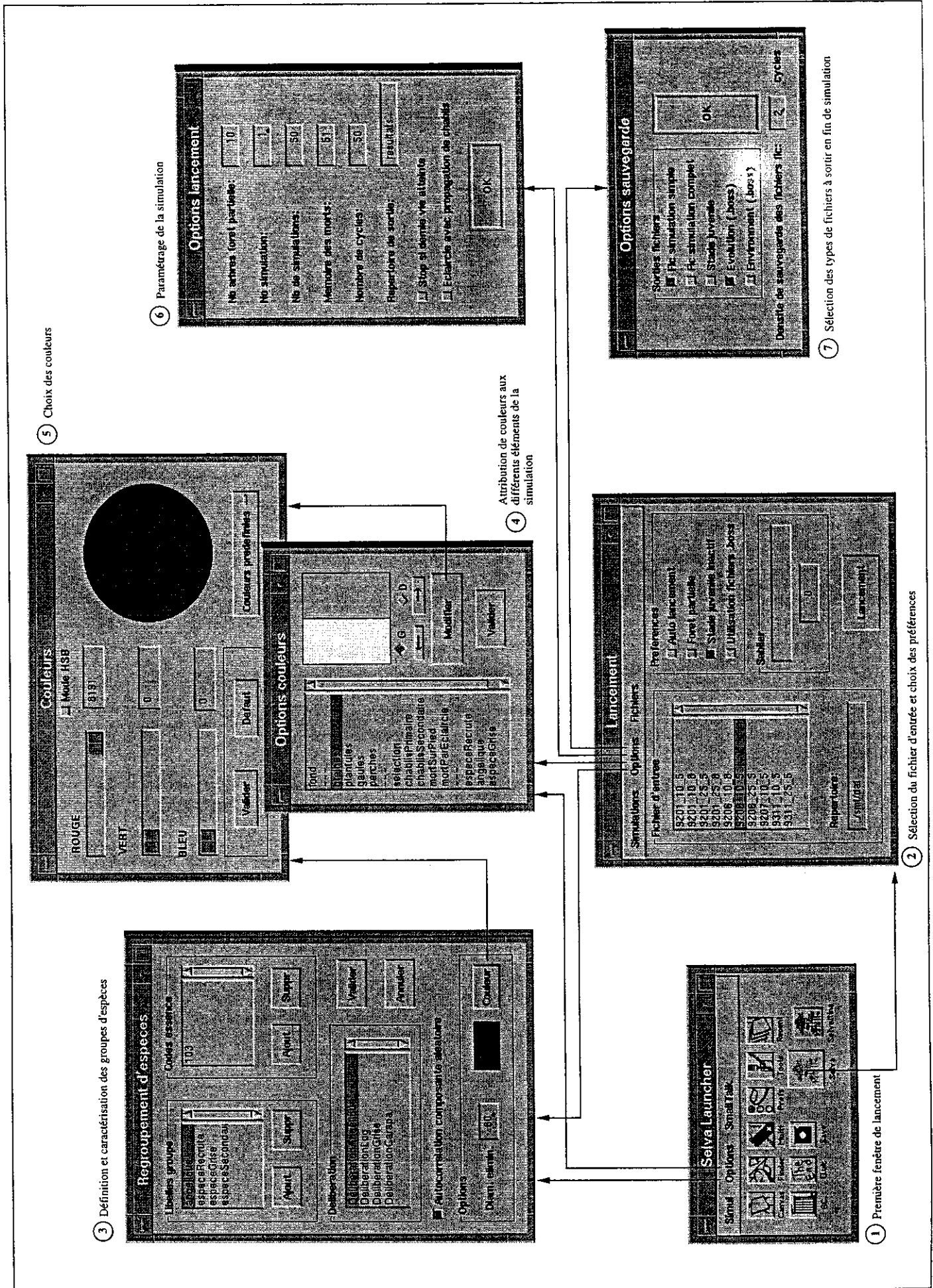
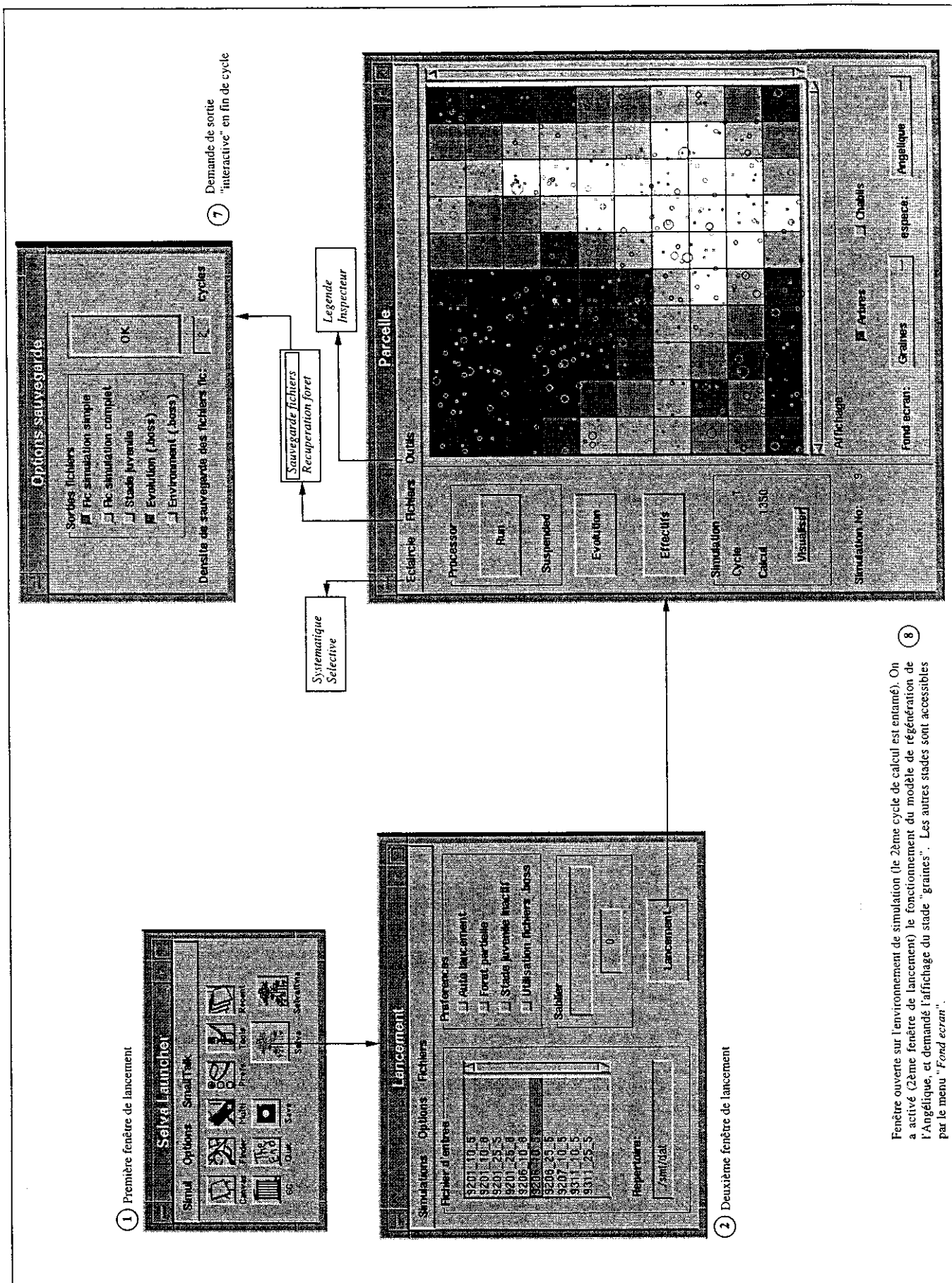


Fig.1. Paramétrage d'une simulation à l'aide de l'interface de SELVA.



Fenêtre ouverte sur l'environnement de simulation (le 2ème cycle de calcul est entamé). On a activé (2ème fenêtre de lancement) le fonctionnement du modèle de régénération de l'Angélique, et demandé l'affichage du stade "graines". Les autres stades sont accessibles par le menu "Fond écran".

Fig.2 - Lancement d'une simulation à partir de l'interface de SELVA.

Une série d'éléments concernant : (i) les différents stades de la régénération, (ii) les différents types de mort, (iii) les sélections réalisées à l'écran à l'aide de la souris et (iv) les EGE, apparaissent dans la liste de gauche. On peut sélectionner ces éléments et leur affecter des couleurs spécifiques par l'intermédiaire de la fenêtre « Couleurs ». Ceci est utile pour la représentation de l'information dans la fenêtre (8).

En ce qui concerne les stades de la régénération, on peut affecter à chacun un nuancier de 10 couleurs (choisies à la main) qui vont représenter différentes intensités d'occupation de chaque placeau. La fenêtre « Parcelle » (8) sur la Fig.2 présente un exemple pour l'Angélique : le fond d'écran est occupé par l'information « graines » disséminées par les reproducteurs. La légende du nuancier peut-être appelée à partir du menu « Outils » de cette fenêtre, item <Légende>.

1.2.3 - Caractéristiques de la simulation

La fenêtre « Options lancement » est ouverte à partir du menu « Options » de la fenêtre « Lancement ». Elle permet à l'utilisateur de fixer :

- le nombre d'individus qu'il souhaite charger à partir des fichiers parcelle, s'il ne veut travailler que sur une partie du peuplement ;
- le nombre de répétitions de simulations qu'il souhaite voir réalisées à partir du même état initial ;
- le nombre de cycles pendant lesquels le système doit garder trace des individus morts ;
- le nombre de cycles à effectuer pendant la simulation ;
- le répertoire où doivent être stockés les résultats ;
- l'arrêt ou non de la simulation lorsque le temps de demi-vie est atteint (temps au bout duquel la moitié de l'effectif vivant initialement présent a disparu) ;
- la propagation ou non de l'effet d'un chablis primaire : la simulation d'une éclaircie par empoisonnement requiert la désactivation de cette option.

1.2.4 - Choix du type et de la configuration des fichiers de sortie

On utilise pour ce faire la fenêtre « Options sauvegarde », ouverte à partir du menu « Options » de la fenêtre « Lancement ». L'utilisateur peut obtenir :

- un fichier simplifié contenant l'identifiant de chaque arbre (carré, numéro, code espèce, abscisse, ordonnée) et la liste des diamètres ;
- un fichier complet recensant également la valeur des indices de compétition ;
- un fichier « Juvéniles » contenant, pour une espèce donnée, l'identifiant de chaque placeau et les stocks de graines, plantules, gaules et perches recensés à chaque cycle ;
- des fichiers binaires (.BOSS) stockant toute la configuration de l'environnement de simulation.

L'enregistrement des informations peut concerner tous les cycles sans exception, ou bien 1 cycle sur n, n étant défini par l'utilisateur.

1.3 - Lancement d'une simulation

Le lancement s'effectue à partir de la fenêtre « Lancement » (voir enchaînement Fig.2). Il nécessite la sélection d'un fichier contenant les informations liées à la parcelle sur laquelle l'utilisateur veut travailler (s'il veut travailler sur une parcelle « vide », il lui suffit d'activer le bouton <Foret partielle> et d'indiquer « 0 » dans le champ « Nb arbres foret partielle » de la fenêtre (6)). Ces fichiers peuvent éventuellement être des fichiers binaires (.BOSS) permettant de recharger un environnement complet précédemment sauvegardé. On peut également choisir de travailler ou non avec le module complet de régénération (qui ralentit les simulations).

L'activation du bouton <Lancement> entraîne l'ouverture en lecture des fichiers et la mise en place de l'environnement, décrite au § 7.2.2.1, chapitre 7 du texte principal. Cette étape se termine par l'ouverture d'une fenêtre sur la parcelle « virtuelle » ((8), voir Fig.2).

2 - Description des classes de SELVA

L'essentiel des classes du simulateur figure sur les organigrammes n°1 et 2, qui présentent le cheminement de l'information d'une classe à l'autre au moment de la mise en place de l'interface et des éléments d'une simulation. La plupart des fenêtres de l'interface se trouve représentée sur les figures 1 et 2, ainsi qu'au § 7.3, chapitre 7 du texte principal. Les styles utilisés dans le texte désignent : Classe, attribut, **méthode**, *message*. >> désigne un enchaînement de messages.

2.1 - Les fenêtres

2.1.1 - ForestLauncher

Voir Fig.1 et Fig.2, la fenêtre intitulée « Selva Launcher ».

Héritée de la fenêtre principale de lancement de SmallTalk-80, elle permet de faire appel à toutes les fonctionnalités de VisualWorks 1.0 (logiciel de programmation utilisé) et contient par ailleurs les icônes de lancement de SELVA et de SELVETTINA.

Le cliquage sur l'icône SELVA active la méthode **selva** : *"FenetreMain new open"*.

Par ailleurs, un menu permet d'avoir accès aux classes :

- Options (une instance de cette classe est créée lors du premier lancement de SELVA — cliquage sur l'icône —, elle est affectée à la variable globale SelvaOptions et elle reste ensuite dans l'image : on la retrouve d'une session à l'autre. Le menu "Options" >> "Save SelvaOptions" permet de sauver l'instance dans un fichier BOSS par appel à Tools : *"SelvaOptions save" >> "Tools save: self onFileWithConfirm: 'SelvaOptions.boss'"*, et "Options" >> "Retraver SelvaOptions" permet de la récupérer : *"Options loadSelvaOptions" >> "SelvaOptions:= Tools loadWithConfirm: 'SelvaOptions.boss'"*. "Simul" >> "Reinitialisation SelvaOptions" crée une nouvelle instance de Options qui écrase la précédente : *"Options initializeSelvaOptions" >> "^SelvaOptions:= self new initialize"*) ;
- Tools (gestion de la mémoire, sauvegarde et chargement de SELVA, sauvegarde et chargement de SelvaOptions, par l'intermédiaire de la classe Options) ;
- FenetreMainEvolution (voir plus bas) ;
- FenetreEspeces (voir plus bas) ;
- FenetreOptionsCouleurs (voir plus bas).

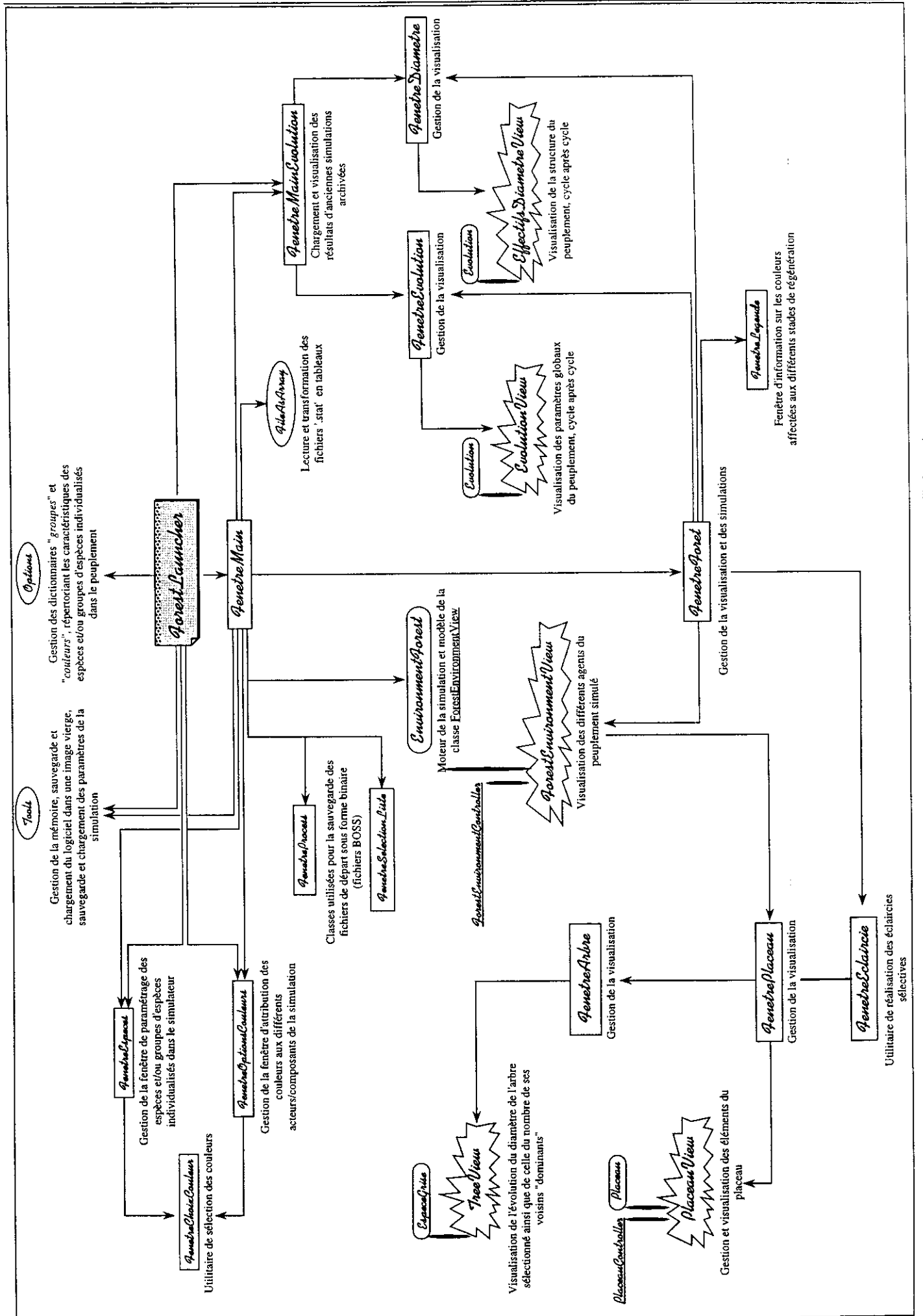
2.1.2 - FenetreMain

Voir Fig.1 et Fig.2 la fenêtre intitulée « Lancement ».

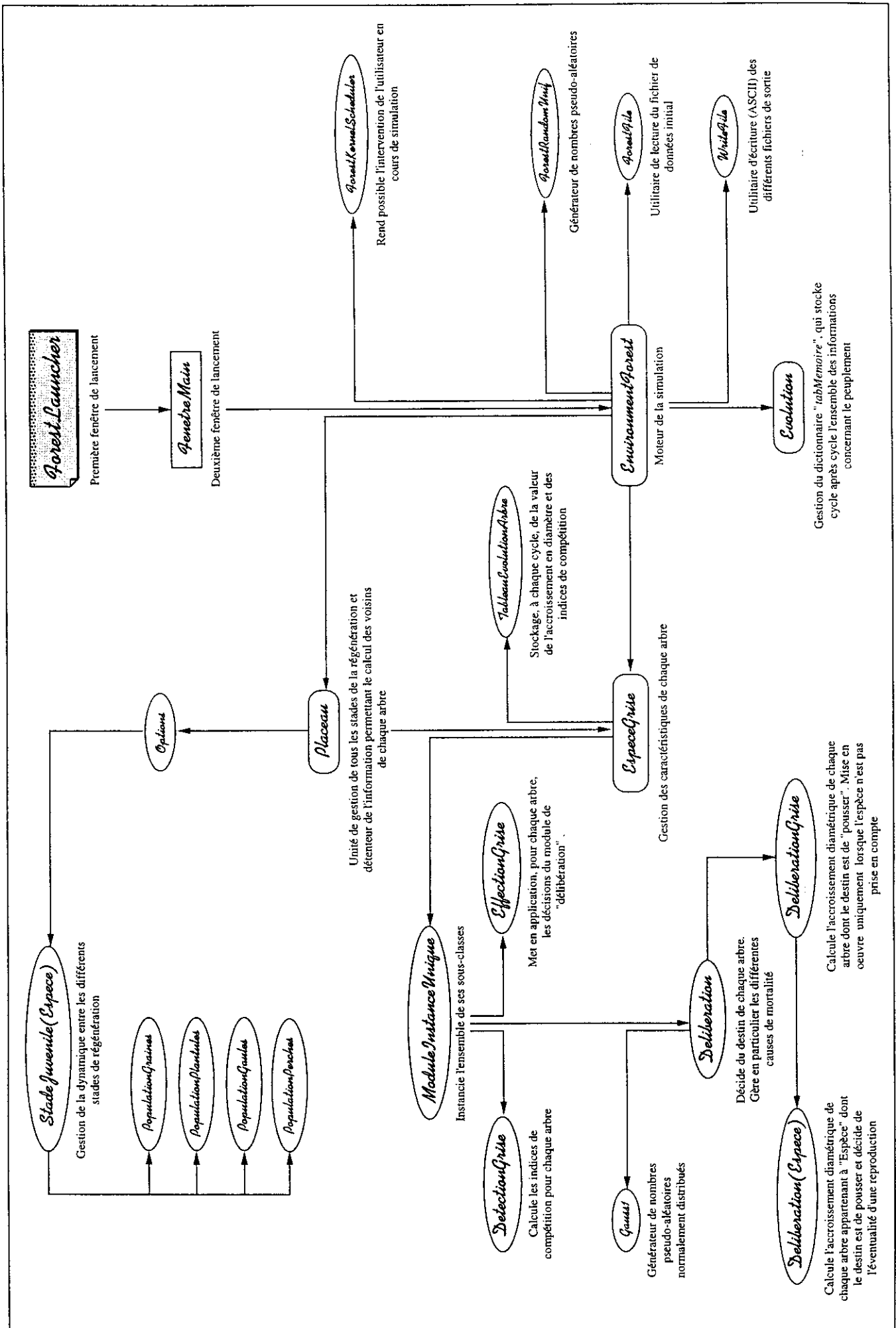
Fenêtre principale du simulateur, qui permet de passer un certain nombre de paramètres nécessaires à la création de l'environnement forestier de départ et de définir les caractéristiques des simulations et de leur sauvegarde.

Le cliquage sur le bouton "Lancement" active la méthode **lancement** qui a pour effet de créer à la fois une instance de FenetreForet (voir plus bas) et de EnvironmentForest (que nous noterons souvent **EF**). Les paramètres nécessaires à la constitution de EnvironmentForest sont passés à ce moment-là.

Un menu permet d'avoir accès aux classes FenetreMainEvolution (voir plus bas) par la méthode **evolution** ("Simulations" >> "Chargement"), FenetreEspeces (voir plus bas) par la méthode **groupes** ("Options" >> "Especes"), FenetreOptionsCouleurs (voir plus bas) par la méthode **optionCouleurs** ("Options" >> "Couleurs"), FenetreProcess (voir plus bas) et FenetreSelectionListe (voir plus bas) par la méthode **sauvegardeFichiersBoss** ("Fichiers" >> "Creation fichiers boss") et Tools, par les méthodes **retraveOptionsConfirm** ("Fichiers" >> "Recuperation Options") et **saveOptions** ("Fichiers" >> "Sauvegarde Options"). Ces deux dernières méthodes permettent de sauvegarder et récupérer tous les paramètres passés à FenetreMain, pour un travail ultérieur.



Organigramme n°1 - Mise en place de l'interface du simulateur. Les flèches indiquent les voies et le sens principaux de circulation de l'information.



Organigramme n°2 - Mise en place et activation des éléments de la simulation. Les flèches indiquent les voies et le sens principaux de circulation de l'information.

A noter également deux autres fenêtres simples appelées par les méthodes **fenetreParametres** ("Options" >> "Parametres") et **fenetreSauvegarde** ("Options" >> "Sauvegardes") : elles ne correspondent pas à des instances de classes (elles permettent le passage de paramètres), mais à des "WindowSpec" particulières du canevas de **FenetreMain**.

Accès enfin à la classe **FileAsArray** (qui permet de lire les infos contenues dans un fichier et de les transformer en tableau : cette classe est utilisée pour le décodage du fichier '.stat' qui contient les deux infos (1) taille d'un côté du monde en nombre de placeaux et (2) n° de la campagne de démarrage dans le fichier) par la méthode **initEnvironmentFic**.

Variables d'instance.

- *autoLancement* : vrai ou faux.

- *compteurSablier* : il s'agit de la variable "attachée" au champ numérique du tableau "Sablier" apparaissant sur le canevas de **FenetreMain**. Cette variable est incrémentée par **EnvironmentForest** au moment de la lecture des fichiers '.vivants' et '.morts', à l'aide du message : "*sablier compteur: cont*" (*sablier* est une variable de **EF** contenant l'instance de **FenetreMain**).

- *cyclesSauve* : nombre total de cycles pour lesquels il faudra stocker des résultats.

- *demieVie* : vrai ou faux.

- *densiteSauvegarde* : stocke le pas de temps n des sauvegardes à réaliser (tous les n cycles).

- *fichierFicComplet* : vrai ou faux (concerne la sortie fichier des résultats). Dans ce cas, on stocke à la fois les valeurs des diamètres et des indices de compétition, cycle après cycle. Variable attachée au champ "*Fic simulation simple*" de la fenêtre "Options">>"Sauvegardes".

- *fichierFicSimple* : vrai ou faux (concerne la sortie fichier des résultats). Seuls les diamètres cycle après cycle sont enregistrés pour chaque arbre.

- *fichierForest* : vrai ou faux. Provoque, en fin de simulation, le stockage de l'instance courante de **EnvironmentForest** sous forme de fichier BOSS. Variable attachée au champ "*Environment (.boss)*" de la fenêtre "Options">>"Sauvegardes".

- *fichierEvolution* : vrai ou faux. Provoque, en fin de simulation, la fabrication d'un fichier ".evolution" qui stocke l'évolution des paramètres globaux de la forêt au cours d'une simulation (fichier BOSS). Rejoué ensuite par **FenetreMainEvolution**. Cette variable est attachée au champ "*Evolution(.boss)*" de la fenêtre "Options">>"Sauvegardes".

- *fichiersBoss* : vrai ou faux (construction de l'environnement à partir d'un fichier boss). Variable attachée au champ "*Utilisation fichiers .boss*" de la fenêtre principale du canevas de **FenetreMain**. Associée à la variable *foretPartielle*, elle renseigne, par l'intermédiaire de la méthode **utiliseFichiersBoss**, la variable *lectureFichiersBoss* de **FenetreForest**. **NB** : il est clair que les variables *fichiersBoss* et *foretPartielle* sont antagonistes : elles ne peuvent être vraies en même temps, ce qui est vérifié par **utiliseFichiersBoss**. Cette méthode interdit l'usage d'un fichier BOSS si *foretPartielle* est vraie.

- *fichierStadeJuvenile* : vrai ou faux. Provoque le stockage sous forme ASCII, en fin de simulation, des résultats concernant la régénération. Variable attachée au champ "*Stade juvenile*" de la fenêtre "Options">>"Sauvegardes".

- *ficList* : type **SelectionInList**. Contient les noms des fichiers ayant une extension ".stat" et une extension ".forest" (fichiers boss), l'extension étant tronquée. Les fichiers ".forest" n'apparaissent que si *fichiersBoss* est vrai.

- *foretPartielle* : vrai ou faux.

- *labelSablier* : variable attachée au champ d'écriture du tableau "Sablier" apparaissant sur le canevas de **FenetreMain**. Elle est renseignée par **EnvironmentForest** au moment de la lecture des fichiers '.vivants' et '.morts', à l'aide du message : "*sablier label: 'lecture fichier ...'*" (*sablier* est une variable de **EF** contenant l'instance de **FenetreMain**).

- *nbArbres* : il s'agit du nombre d'arbres à charger lorsque *foretPartielle* est vrai. Ce paramètre est passé à EnvironmentForest au moment de son initialisation (`initEnvironmentFic: <>` et `initEnvironmentBoss`). Il vaut 0 si *foretPartielle* est faux.

- *nbDeSimulations* : nombre de simulations à réaliser.

- *nbMaxCycles* : renseigne la variable *cyclesAEffectuer* de EnvironmentForest. Nombre total de cycles à effectuer par la simulation.

- *nomRepertoire* : nom du répertoire contenant les fichiers de données de départ (y compris fichiers boss s'ils existent).

- *noSimul* : renseignée à partir de la valeur de la variable de classe *NoSimulation* au moment de l'instanciation de FenetreMain. Elle renseigne à son tour cette variable de classe au moment d'un nouveau cliquage sur le bouton "Lancement". **NB** : elle est attachée au champ "No simulation" de la fenêtre "Options" >> "Parametres", donc on peut lui attribuer une valeur en début de travail.

- *noSimulationCourante* : remise à zéro à chaque création d'instance de FenetreMain. Cette variable est incrémentée à chaque "cliquage" sur le bouton "Lancement" : `"lancement: icone" >> "noSimulationCourante:= noSimulationCourante+ 1"`. Lorsque l'instance courante de EnvironmentForest est parvenue au bout de son nombre imposé de cycles, elle lance le message : `"self changed: #terminate"`. Ce message est récupéré par ForestEnvironmentView (que nous appellerons aussi FEV), qui l'interprète de la manière suivante : `"(anAspect = #terminate) ifTrue: [self finDeLaSimulation. ^self]"` (méthode `update: anAspect with: anObject`. FEV demande alors à FenetreMain : `relancerSimulation`. Cette méthode compare *noSimulationCourante* à *nbDeSimulations* et renvoie un booléen à FEV, qui décide éventuellement de recharger une nouvelle instance de EnvironmentForest à partir d'un fichier BOSS '.forest'.

- *repertoireSortie* : stocke le nom du répertoire dans lequel seront écrits les fichiers résultat.

- *stadeJuvenileInactif* : vrai ou faux. Variable attachée au champ "Stade juvenile inactif" du canevas de FenetreMain.

Variables de classe.

- *NoSimulation* : conserve le numéro des simulations d'une simulation sur l'autre. Remis à zéro quand on quitte SELVA. Cette variable est incrémentée lors de chaque cliquage sur le bouton "Lancement" : `"lancement: icone" >> "NoSimulation:= noSimul + 1"`.

2.1.3 - FenetreForest

Voir Fig.2 la fenêtre intitulée « Parcelle » ; voir également le chapitre 7, paragraphe 7.3 du texte principal.

Cette fenêtre a pour mère FenetreMain. C'est elle-même une fenêtre mère. Elle permet de lancer les simulations et de les suivre en temps réel (boutons d'activation, d'interruption, menu permettant de réaliser des éclaircies, inspection des placeaux, suivi des paramètres du peuplement, examen des arbres individuels, etc.). Elle contient une vue sur EnvironmentForest, ie elle crée, au moment de son initialisation, une instance de ForestEnvironmentView dont le modèle est l'instance de EnvironmentForest, précédemment créée par FenetreMain.

La souris, lorsqu'elle se déplace sur la vue ForestEnvironmentView, acquiert des propriétés particulières gérées par la classe ForestEnvironmentController. Il faut d'abord (c'est indispensable) sélectionner un placeau en cliquant dessus avec le bouton de gauche (récupération de coordonnées), puis le bouton central donne accès à un menu permettant notamment la création d'une instance de la classe FenetrePlaceau (voir plus bas) par l'article "Zoom Placeau".

Un menu permet d'activer les classes : FenetreEclaircie (voir plus bas) par la méthode `eclaircieSelective` (`eclaircieSystematique` renvoie à une méthode de EnvironmentForest), FenetreLegende (voir plus bas) par la méthode `fenetreLegende`. Par ailleurs, deux boutons permettent d'accéder au suivi des paramètres du peuplement (revoir les intitulés) : accès à FenetreDiametre par la méthode `effectifsDiametre`, et à FenetreEvolution par la méthode `evolutionAct`.

Dans le même menu, on a accès à la méthode **fenetreSauvegarde** de FenetreMain (voir plus haut).

Les boutons "run", "suspend", "terminate", agissent sur le scheduler (instance de ForestKernelScheduler) de EnvironmentForest.

Variables d'instance.

- *agentNumber* : stocke le numéro de l'arbre en cours de traitement dans un cycle donné. Remis à 0 au moment du passage au cycle suivant.
- *arbresFlag* : vrai ou faux. Commande l'affichage des individus "arbres" sur la carte.
- *centSimulations* : vrai ou faux. Cette valeur est attribuée par FenetreMain au moment du cliquage sur l'icône "Lancement", si sa variable *nbDeSimulations* est supérieure à 1. La variable *centSimulations* est testée par FenetreForet dans sa méthode **postOpenWith: aBuilder**. Si elle est à "true" le message **saveEnvironmentForestIfNeed** est envoyé à EnvironmentForest. Il a pour effet de provoquer une sauvegarde de l'instance courante **initiale** (avant démarrage de la simulation !) de cette classe sous forme de fichier BOSS, en vue de sa réutilisation pour les répétitions suivantes.
- *chablisFlag* : vrai ou faux. Commande l'affichage des chablis primaires et secondaires sur la carte.
- *cycle* : stocke le numéro du cycle en cours de traitement. Initialisé à 0, au moment de l'ouverture de FenetreForet.
- *lectureFichierBoss* : vrai ou faux. Cette valeur est attribuée au moment du cliquage sur l'icône "Lancement" de FenetreMain. On a : "lancement: icone" >> "fenetre lectureFichierBoss: self utiliseFichiersBoss". La méthode **utiliseFichiersBoss** renvoie "vrai" ou "faux" en fonction de la valeur simultanée des deux variables booléennes *fichiersBoss* et *foretPartielle* de FenetreMain. Ensuite, la valeur de *lectureFichierBoss* est testée par **lireForetInFichier** de EnvironmentForest.
- *menuFondHolder* : contient une étiquette caractérisant la couleur du fond de carte. Par défaut : #fondUni asValue, ie cela correspond à un fond de carte "vide de régénération". Autres valeurs possibles : #graines, #plantules, #gaules, #perches (voir § 2.5).
- *menuEspeceHolder* : contient le nom de l'espèce dont la régénération est affichable sur la carte. C'est "#angelique asValue", par défaut. Le menu utilisé est le même que celui de FenetrePlaceau (voir plus bas). Toute action sur le menu provoque l'envoi du message : **changeMenuEspèces** à "self". Ce dernier demande "vue update". La méthode de classe **menuEspece** appelle la méthode du même nom de la classe FenetrePlaceau.
- *nbArbresFlag* : vrai ou faux. Commande l'affichage sur la carte de cercles de rayon croissant avec le nombre d'arbres présents dans les différents placeaux de 10mx10m.
- *nbMortsFlag* : vrai ou faux. Commande l'affichage sur la carte de cercles de rayon croissant avec le nombre total d'arbres morts sur les différents placeaux (10mx10m).
- *noSimulation* : stocke le numéro de la simulation en cours. Initialisé à 1 dans la méthode **initFlags**. Lors de son incrémentation (quand les relancements sont automatiques), passe sa valeur à la variable *noSimulationCourante* de FenetreMain.
- *surfaceTerriereFlag* : vrai ou faux. Commande l'affichage sur la carte de cercles de rayon croissant avec l'importance de la surface terrière calculée sur les différents placeaux (10mx10m).
- *visible* : vrai ou faux. Valeur modifiable par le bouton de visualisation, qui active la méthode **visualiser** : elle permet de rendre invisible la carte et un certain nombre de boutons, dans le but d'accélérer la vitesse de traitement de la simulation.
- *vue* : représente l'instance de ForestEnvironmentView, créée par la méthode **initializeOn: anEnvironment**: "vue:= ForestEnvironmentView new model: anEnvironment".

2.1.4 - FenetreMainEvolution

Cette fenêtre a pour mère FenetreMain : elle est appelée à partir de son menu (méthode **evolution**), mais elle peut également être ouverte à partir du menu de ForestLancer (méthode **optionsMenu**).

Cette classe crée une instance de Evolution (dans la méthode **lancement**). Elle crée par ailleurs, à la demande, une instance de FenetreDiametre (voir plus bas) ou de FenetreEvolution (voir plus bas).

Elle permet en fait de visualiser (de rejouer) les résultats d'anciennes simulations, si leur sauvegarde a été demandée (sous forme de fichiers ".evolution").

Variables d'instance.

- *evolution* : stocke les informations concernant l'évolution des paramètres de la forêt au cours d'une simulation. Dans la méthode **lancement**: **repertoireSortie** activée par **evolution** (de FenetreMain), on trouve : "*evolution:= Evolution load: (liste at: indice)*". « Liste » contient tous les noms de fichiers ".evolution" (contenus dans repertoireSortie).

- *nomFichier* : stocke le nom du fichier présent dans "*liste at: indice*" (voir précédemment).

2.1.5 - FenetreProcess

A pour fenêtre mère FenetreMain. Cette fenêtre gère ici la fabrication des fichiers boss (d'une manière plus générale, elle permet d'activer et de désactiver n'importe quel process contenant des "Processor yield" dans son code). FenetreMain passe à cette fenêtre, par l'intermédiaire de la méthode **sauvegardeFichiersBoss** un process qui est le suivant : "*process:= self processSauvegardeFichiersBoss: fenetre*". Ce process contient des "Processor yield", et crée des environnements sauvegardés sous forme de fichiers boss.

Deux boutons essentiels : "*run*" et "*stop*". A l'ouverture de la fenêtre, seul le bouton "*run*" est visible.

Variables d'instance.

- *label* : contient une chaîne de caractères passée par la méthode **processSauvegardeFichiersBoss** (de FenetreMain) : "Traitement fichier" + le nom du fichier en cours de traitement s'affichent en tête de la fenêtre FenetreProcess.

- *process* : contient le process passé par la méthode ci-dessus.

2.1.6 - FenetreSelectionListe

A pour fenêtre mère FenetreMain. Une instance est créée par la méthode **processSauvegardeFichiersBoss: fenetre**, le paramètre passé étant une instance de FenetreProcess. D'une manière générale, cette classe permet de gérer deux listes de fichiers : la sélection d'un fichier dans une liste provoque son transfert dans l'autre et vice-versa. Ici, ce système est utilisé pour choisir les fichiers qui doivent être transformés en fichiers boss. Une fois le choix (réversible donc) fait par l'utilisateur, la liste des fichiers choisis ainsi constituée entre dans un process de traitement systématique passé à FenetreProcess, qui permet une interruption à tout moment !

Variables d'instance.

- *commentaire* : contient une chaîne de caractères passée par la méthode **processSauvegardeFichiersBoss: fenetre** de FenetreMain.: "Sélectionner les fichiers à sauvegarder sous forme de fichiers BOSS".

- *listeRefuses* : contient la liste originelle, puis la liste des fichiers non sélectionnés.

- *listeResultat* : contient la liste définitive des fichiers sélectionnés.

- *listeSelection* : contient la liste en cours des fichiers sélectionnés.

2.1.7 - FenetreEspeces

Voir Fig.1 la fenêtre intitulée « Regroupement d'especes ».

A pour fenêtre mère FenetreMain, mais une instance de cette fenêtre peut également être créée au niveau du menu de ForestLancher. Elle permet la saisie de l'intitulé des espèces et/ou groupes d'espèces pris en compte, et la définition de leurs caractéristiques pour la simulation. Ces caractéristiques apparaissent par ailleurs automatiquement dès que l'on sélectionne un intitulé de la liste (récupération des infos précédemment stockées dans la variable globale SelvaOptions, par l'intermédiaire de la variable dicoGroupes).

Pour saisir l'information "couleur d'affichage" affectée à chaque groupe (clé "#couleur" de *dicoGroupes*), la fenêtre crée une instance de FenetreChoixCouleur (voir plus bas, et illustration Fig.1) par l'intermédiaire de la méthode **modifCouleur**. Lors de cette création, la couleur présente par défaut dans *dicoGroupes* est passée à FenetreChoixCouleur qui l'affiche. En retour, lorsque la méthode **valider** de FenetreChoixCouleur est activée, elle passe à FenetreEspeces un tableau contenant la nouvelle couleur choisie (tab at: 1) et le libellé correspondant (tab at: 2), permettant de renseigner *dicoGroupes* (méthode **validationColor: tab**) et de dessiner dans un cartouche *ad hoc* la couleur choisie pour le groupe créé.

NB. Lorsque l'on crée un nouveau libellé d'espèce ou de groupe d'espèces, les champs de *dicoGroupes* sont automatiquement remplis par des valeurs par défaut : "*dicoGroupes at: dernierVenu put: (Options defaultEspece)*" (méthode **ajoutLibelle**).

Les classes correspondent aux intitulés des modules de délibération associés à chaque groupe d'espèces.

Variables d'instance.

- *diametreElimination* : stocke la valeur du diamètre au-delà de laquelle l'arbre sera éliminé en cas d'éclaircie systématique ou d'exploitation forestière. Si l'espèce doit être préservée, il suffit d'indiquer une valeur suffisamment élevée (1000).

- *dicoGroupes* : dictionnaire géré par Options (SelvaOptions), sauvegardé de manière permanente et contenant les champs suivants : *libellé de l'espèce ou du groupe d'espèces / code(s) essence correspondant(s) / couleur d'affichage / classe (?) / diamètre d'élimination / autocorrélation*. A chaque création d'une instance de FenetreEspeces, les infos contenues par SelvaOptions sont rapatriées dans *dicoGroupes* et dès qu'une validation des nouveaux choix intervient, *dicoGroupes* est retransféré dans SelvaOptions.

- *listeClasses* : liste constituée de la manière suivante. On demande à la classe EspeceGrise de retourner toutes les sous-classes de la super-classe Deliberation, puis d'en éliminer les sous-classes DeliberationPetitGris et DeliberationEclaircie. Tous les noms restant sont transformés en symboles qui apparaissent dans la liste du cadre "Deliberation" de la fenêtre gérée par FenetreEspeces.

- *listeCodeEssence* : stocke le(s) numéro(s) de code essence rattaché(s) à un libellé (espèce ou groupe d'espèces) (voir méthodes **ajoutCodeEssence** et **supprimerCodeEssence**).

- *listeLibelles* : stocke la liste des libellés saisis (voir méthodes **ajoutLibelles** et **supprimerLibelle**).

- *randomAutocorrele* : vrai ou faux. Indique si l'on veut introduire une autocorrélation entre résidus successifs du modèle de croissance.

2.1.8 - FenetreOptionsCouleurs

Voir Fig.1 la fenêtre intitulée « Options couleurs ».

A pour fenêtre mère FenetreMain, mais une instance de cette fenêtre peut également être créée au niveau du menu de ForestLauncher.

Cette fenêtre est le pendant de FenetreEspeces (fonctionne à peu près de la même manière), mais elle concerne uniquement l'attribution de couleurs à différents objets qui sont soit des espèces soit des éléments comme : fond d'écran, graines, plantules, chablis primaires, chablis secondaires, arbres sélectionnés, etc. Une liste de base est initialisée : #fond, #graines, #plantules ... (voir *listeObjets*, ci-dessous), à laquelle on rajoute les infos permanentes contenues dans *dicoCouleurs* (même principe que *dicoGroupes*), par l'accès à la classe Options (et SelvaOptions). On trouve dans *dicoCouleurs* des couleurs par défaut que l'on peut modifier : cela se fait par l'appel à une instance de FenetreChoixCouleur (méthode **modifierCouleur**). En retour, lorsque la méthode **valider** de FenetreChoixCouleur est activée, elle passe à FenetreOptionsCouleurs un tableau contenant le libellé concerné et la, ou les couleurs choisies (méthode **validationColor: tab**), permettant de renseigner *dicoCouleurs* et de dessiner dans un (des) cartouche(s) *ad hoc* la (les) couleur(s) choisie(s) pour l'objet concerné.

La difficulté principale réside dans le fait que lorsque les objets sont : #graines, #plantules, #gaules ou #perches, ce n'est pas une mais 10 couleurs qui sont concernées : elles représentent en effet 10 classes de densité possibles. Deux cartouches apparaissent donc, concernant deux valeurs "adjacentes" de densité et il faut les balayer de 1 à 10. D'où une gestion un peu compliquée de flèches gauche et droite, sans compter le choix des 10 couleurs.

Variables d'instance.

- *dicoCouleurs* : dictionnaire géré par Options (SelvaOptions), sauvegardé de manière permanente et contenant les champs suivants : *objet / couleur*. A chaque création d'une instance de FenetreOptionsCouleurs, les infos contenues par SelvaOptions sont rapatriées dans *dicoCouleurs*, et dès qu'une validation des nouveaux choix intervient, *dicoCouleurs* est retransféré dans SelvaOptions.

- *gaucheDroite* : contient #gauche, ou #droite. Permet apparemment le calcul d'un indice de positionnement.

- *indiceDebut* : initialisé à 1. Permet le calcul d'une position entre 1 et 10 pour les stades de régénération.

- *listeObjets* : contient les libellés suivants : #fond, #graines, #plantules, #gaules, #perches, #selection, #chablisPrimaire, #chablisSecondaire, #mortSurPied, #mortParEclaircie. On y ajoute ensuite les autres éléments contenus dans *dicoCouleurs*.

2.1.9 - FenetreChoixCouleur

Voir Fig.1 la fenêtre intitulée « Couleurs ».

A pour fenêtres mères soit FenetreOptionsCouleurs, soit FenetreEspeces. Les instances de ces classes passent à FenetreChoixCouleur un libellé (voir *identifiantColor*) et une couleur par défaut (voir *defaultColor*) qui est affichée. L'affichage par défaut se fait selon le mode RVB (le mode HSB peut être activé par l'utilisateur). Le choix d'une autre couleur par action sur le RVB fait appel à une classe ColorValue. La méthode **valider** renvoie à la fenêtre appelante un tableau contenant un code de couleur et le libellé *identifiantColor*.

NB. On peut accéder à un menu de couleurs par défaut.

Variables d'instance.

- *blueValue* : composante bleue de la couleur résultante.

- *brightValue* : brillance de la couleur résultante.

- *couleurPredefini* : ??? Initialisée à nil.

- *default color* : couleur stockée par défaut dans les dicos *dicoCouleurs* et *dicoGroupes* (voir SelvaOptions), et passée à FenetreChoixCouleur par les fenêtres appelantes.

- *greenValue* : composante verte de la couleur résultante.

- *hueValue* : intensité de la couleur résultante.

- *identifiantColor* : libellé stocké dans les dicos *dicoCouleurs* et *dicoGroupes* (voir SelvaOptions), et passé à FenetreChoixCouleur par les fenêtres appelantes.

- *modeHSB* : vrai ou faux. Permet de basculer d'une représentation RVB à une représentation HSB.

- *redValue* : composante rouge de la couleur résultante.

- *satValue* : saturation de la couleur résultante.

2.1.10 - FenetrePlaceau

Voir chapitre 7, § 7.3 du texte principal.

Fenêtre ouverte lorsque l'on utilise la souris sur un placeau apparaissant sur la carte affichée par FenetreForet. L'ouverture est commandée par ForestEnvironmentController (item "Zoom Placeau" du menu de la souris).

L'initialisation de l'instance de FenetrePlaceau se fait par la méthode **initialize: lePlaceau**, donc par passage d'un argument qui est une instance de Placeau. Cette instance contient un certain nombre d'infos qui sont récupérées par FenetrePlaceau (liste des arbres présents : voir *listeArbres*, effectifs des graines, plantules, gaules, perches, etc.).

Une instance de la classe PlaceauView (dont le modèle est Placeau) est créée par la méthode **initialize: lePlaceau**. On peut sélectionner les arbres qui apparaissent, ou les désélectionner, soit en utilisant la liste soit en utilisant la carte. La gestion de 4 flèches (méthodes **flecheBasAction / flecheDroiteAction / flecheGaucheAction / flecheHautAction**) permet de se déplacer d'un placeau à l'autre.

Sur un arbre sélectionné, on peut activer un **inspect** (méthode **inspection**) et on a accès à la classe **FenetreArbre** (voir plus bas) par la méthode **evolutionArbre**.

Variables d'instance.

- *graines* : stocke les effectifs de graines présents sur le plateau (info. contenue dans l'instance de **Placeau** et rapatriée par la méthode **initializeStadesJuveniles**).
- *listeArbres* : à l'initialisation de **FenetrePlaceau**, la méthode **initializeListeArbres** est activée. Elle permet de rapatrier dans *listeArbres* les infos contenues dans la variable *listeArbres* de **Placeau** : "**listeArbres:=placeauView placeau listeArbres copy**".
- *listeNomsArbres* : cette liste "double" *listeArbres*. C'est sur elle que l'on agit avec la souris : le cliquage mémorise une position (= un index) qui permet d'atteindre l'arbre (= l'agent) contenu dans *listeArbres* à la position indiquée. La méthode **initialize: lePlaceau** contient l'instruction : "**listeNomsArbres:=SelectionInList new**".
- *menuEspeceHolder* : stocke le nom de l'espèce pour laquelle existent des infos concernant la régénération. Par défaut, cette variable vaut "**#angelique asValue**". Tout changement (saisie d'un autre nom ou cliquage) provoque l'activation du message : **initializeStadesJuveniles**.
- *nomPlaceau* : récupère le nom de l'instance de **Placeau**.
- *oldSelection* : contient l'agent arbre précédemment sélectionné sur la carte, lorsque l'on change de sélection.
- *placeauView* : stocke l'instance de **PlaceauView**.
- *plantules* : stocke les effectifs de plantules présents sur le plateau (info. contenue dans l'instance de **Placeau**, et rapatriée par la méthode **initializeStadesJuveniles**). Voir § 2.5.
- *gaules* : stocke les effectifs de gaules présents sur le plateau ("). Voir § 2.5.
- *perches* : stocke les effectifs de perches présents sur le plateau ("). Voir § 2.5.

2.1.11 - **FenetreEclaircie**

Voir chapitre 7, § 7.3 du texte principal.

A pour fenêtre mère **FenetreForet**. Cette fenêtre hérite de **FenetrePlaceau**. Elle s'initialise sur le premier plateau situé en bas à gauche sur la carte de la parcelle étudiée.

Cette fenêtre communique avec **EnvironmentForest**, notamment lors de l'activation de la méthode **lancerEclaircie** (activation de la méthode **eclaircieSelective: nbArbres** de **EnvironmentForest**). Elle permet la sélection / désélection des arbres à éliminer lors d'une éclaircie sélective. Une annulation générale des choix est possible.

NB. Lorsque l'on clique sur un arbre, une vérification de son état est réalisée. Si "**selectTree etat = #vivant**", le compteur *nbArbresSelectionnes* est incrémenté. Dans le cas contraire (en fait, cas où l'arbre a été précédemment sélectionné pour l'éclaircie sélective), le compteur est décrémenté : cela permet d'annuler un choix précis.

Variables d'instance.

- *nbArbresSelecHectare* : renseignée par le calcul du quotient : *nbArbresSelectionnes / "self environmentForest hectares"* (méthode **changeNbArbres**).
- *nbArbresSelectionnes* : compteur du nombre d'arbres sur lesquels on a cliqué pour les sélectionner.

2.1.12 - **FenetreDiametre**

Voir chapitre 7, § 7.3 du texte principal.

A pour fenêtres mères **FenetreForet** ou **FenetreMainEvolution** (l'ouverture est déclenchée par l'action sur le bouton « diamètre »). Cette fenêtre est constituée de 3 sous-vues, qui sont des instances de : **EffectifsDiametreView**. Ces 3 sous-vues ont pour modèle une instance de **Evolution**.

Les 3 sous-vues présentent des distributions par classes de diamètre : à l'ouverture, elles concernent les effectifs vivants, le total des chablis et les morts sur pied. Par défaut, ce sont toujours les infos du dernier cycle qui s'affichent ("*dernierCycle:= true asValue*").

Au moment de l'initialisation, une instance de EnvironmentForest est passée par FenetreForet à FenetreDiametre qui s'en rend dépendante (cela lui permet d'accéder à l'instance d'Evolution attachée à EnvironmentForest). Au fur et à mesure que l'environnement change (à chaque cycle !), et si la variable *dernierCycle* est activée, le message **update: anAspect with: aParametre** provoque le réaffichage des infos du nouveau cycle par l'intermédiaire de la méthode **cycleChange**. Cette méthode appelle en fait la méthode du même nom appartenant à EffectifsDiametreView. La même méthode est activée lorsque l'on agit sur les flèches droite et gauche de FenetreDiametre (cas où *dernierCycle* vaut "faux"), avec *cycle* incrémenté ou décrémenté de 1 respectivement.

3 menus sont accessibles dans la fenêtre (un par vue !) : ils permettent l'affichage des vivants, des morts totaux, des morts sur pied, des chablis totaux, des chablis primaires et des chablis secondaires.

Variables d'instance.

- *cycle* : stocke la valeur de la variable *cycle* de EnvironmentForest.
- *dernierCycle* : vrai ou faux. Indique si l'on veut voir s'afficher le dernier cycle. Si on désactive le bouton correspondant, il ne se passe plus rien : il faut saisir un numéro de cycle pour voir s'afficher les informations.
- *forest* : stocke une instance de EnvironmentForest, passée par FenetreForet au moment de l'initialisation (de l'ouverture) de FenetreDiametre. On a donc accès à toutes les méthodes de cette classe.
- *view1* : stocke une instance de EffectifsDiametreView.
- *view2* : stocke une instance de EffectifsDiametreView.
- *view3* : stocke une instance de EffectifsDiametreView.

2.1.13 - FenetreEvolution

Voir chapitre 7, § 7.3 du texte principal.

A pour fenêtres mères FenetreForet ou FenetreMainEvolution (l'ouverture est déclenchée par l'action sur le bouton "évolution"). Il s'agit d'une sous-classe de FenetreAvecTableau.

Cette fenêtre montre l'évolution des paramètres globaux de la forêt, cycle après cycle. Elle fait donc appel à une instance de la classe EvolutionView, dont le modèle est une instance de la classe Evolution. Cette instance est passée par FenetreForet au moment de son appel, et FenetreEvolution s'en rend dépendante.

Deux flèches permettent de se déplacer d'un **pas de 20 cycles** (en avant ou en arrière) lorsqu'on les active. Par ailleurs, un menu permet de choisir le type d'info. souhaitée : Nombre d'arbres vivants / Recrutement / Surface terrière / Total des morts / Morts sur pied / Total des chablis / Chablis primaires / Chablis secondaires. Lorsque l'on change de sélection (la valeur de *menuEvolutionHolder* change), un message (**visuChanged: (menuEvolutionHolder value)**) est envoyé à l'instance de EvolutionView, qui se met à jour.

Variables d'instance.

- *arrayView* : stocke l'instance de la classe EvolutionView, créée au moment de l'ouverture de la fenêtre.
- *menuEvolutionHolder* : stocke, me semble-t-il, une étiquette sélectionnée sur le menu disponible dans la vue.

2.1.14 - FenetreArbre

Voir chapitre 7, § 7.3 du texte principal.

A pour fenêtre mère FenetrePlaceau. Il s'agit d'une sous-classe de FenetreAvecTableau, héritant des méthodes **flecheDroite** et **flecheGauche** de la super-classe (déplacement d'un nombre donné de cycles).

L'ouverture se fait par action sur le bouton "Evolution" de FenetrePlaceau. Au moment de l'appel, FenetrePlaceau passe à FenetreArbre l'identité d'un agent arbre (instance de la classe EspecGrise). FenetreArbre fait alors appel à deux instances de TreeView (qui ont pour modèle EspecGrise) : elles permettent d'avoir accès à l'évolution, cycle par cycle, du diamètre de l'arbre d'une part et du nombre de ses voisins (de taille supérieure) d'autre part (méthode

initialize: arbreModel).

Deux flèches (voir la super-classe) permettent de se déplacer d'un pas de 14 cycles (en avant ou en arrière) lorsqu'on les active.

Variables d'instance.

- *diametreView* : stocke une instance de TreeView, qui affiche #diametre.
- *voisinsView* : stocke une instance de TreeView, qui affiche #voisins.

2.1.15 - FenetreLegende

A pour fenêtre mère FenetreForet (par l'intermédiaire d'un menu). Cette fenêtre fait apparaître 10 cartouches colorés correspondant à des niveaux de densité de population aux différents stades de la régénération.

Variables d'instance.

- *color1* : correspond à tabCouleur at: 1.
- *color2* : correspond à tabCouleur at: 2.
- *color3* : correspond à tabCouleur at: 3.
- *color4* : correspond à tabCouleur at: 4.
- *color5* : correspond à tabCouleur at: 5.
- *color6* : correspond à tabCouleur at: 6.
- *color7* : correspond à tabCouleur at: 7.
- *color8* : correspond à tabCouleur at: 8.
- *color9* : correspond à tabCouleur at: 9.
- *tabCouleur* : correspond à un tableau passé par FenetreForet au moment de l'appel à FenetreLegende. Ce tableau est : "*PopulationEffectifsIndividuels tabCouleurs*".

2.2 - Les MVC (Model / View / Controller)

Ce sont les fenêtres (voir paragraphe 1 précédent) qui appellent les vues et communiquent avec elles. Au niveau de la fenêtre, on organise les éléments de visualisation (dont l'emplacement des vues, les flèches de déplacement, etc.) et de dialogue. Au niveau des vues, on gère le graphisme : construction pratique des histogrammes, affichage de rectangles, de carrés, de cercles, etc. C'est à ce niveau également qu'est résolu le problème de l'origine des axes : il faut inverser l'axe des y à l'affichage.

2.2.1 - ForestEnvironmentView

Cette classe aurait dû dériver de GenericEnvironmentView mais D. Caruso a dû changer d'avis. Elle dérive donc directement de AutoScrollingView, et la plupart des méthodes de GenericEnvironmentView sont intégralement reprises.

Une instance est créée par FenetreForet grâce à la méthode **initializeOn: anEnvironment** : "*vue:= ForestEnvironmentView new model: anEnvironment*" (NB : cette méthode est appelée par FenetreMain).

Au niveau de cette classe sont définies des méthodes permettant :

- d'afficher un rectangle ayant des dimensions et une couleur précises (qui peut être "*SelvaOptions couleurs at: #fond*") ;
- d'afficher un cercle centré sur un point précis ;
- d'afficher une ligne, en tenant compte du fait que l'on est dans un univers toroïdal ;
- de convertir les dimensions de l'univers réel en pixels et vice-versa (1 cm = 4 pixels) et de tenir compte de l'inversion de l'axe des y ;

- d'établir la taille "scrollable" de la vue, en partant de l'origine et en utilisant la taille de l'univers étudié converti en pixels (méthode **preferredBounds**).

Méthodes très importantes :

- **displayOn: aGraphicsContext.** Elle détermine la taille du rectangle correspondant à la fenêtre qui a été ouverte, puis elle demande au modèle de lui fournir les "cells" (placeaux) situées à l'intérieur de ce rectangle (après conversion des pixels en cm), avant de les afficher : méthode suivante.
 - **displayCell: aCell.** Cette méthode communique avec FenetreForet. Elle affiche le placeau avec la couleur de fond correspondant au choix qui a été fait d'afficher "Vide", "Graines", "Plantules", "Gaules" ou "Perches" (voir méthode **fondValeur** de FenetreForet et variable *menuFondHolder*) et avec une info "espèce" correspondant à la valeur de *menuEspeceHolder* de FenetreForet. L'info. affichée correspond à celle qui a été sélectionnée par l'utilisateur au niveau des boutons de FenetreForet : voir la valeur (vrai/faux) des variables *arbresFlag*, *chablisFlag*, *surfaceTerriereFlag*, *nbArbresFlag*, *nbMortsFlag* (seul *arbresFlag* est vrai par défaut au départ). Une action sur les boutons correspondants dans FenetreForet active une méthode de ForestEnvironmentView qui change les valeurs de ces variables booléennes en leur opposé.
 - **zoomSelectedPlace.** Cette méthode est appelée par la méthode **zoomPlaceau** de ForestEnvironmentController (voir plus bas). Cette dernière est activée par l'utilisateur à travers le menu de la souris qui apparaît sur la carte une fois qu'un placeau a été sélectionné. **zoomSelectedPlace** a pour effet de créer une instance de FenetrePlaceau et de l'attacher à FenetreForet.
 - **finDeLaSimulation.** Vérifie si des simulations en série ont été spécifiées par l'utilisateur au niveau de FenetreMain, puis communique le résultat à FenetreForet (message **finDeLaSimulation**, ou **nouvelleSimulation**) et charge si nécessaire un nouvel environnement.
 - **loadNewEnvironmentForest: file.** Le fichier concerné est un fichier de type ".forest" (fichier boss). Toutes les sous-fenêtres précédemment ouvertes de FenetreForet sont fermées, une nouvelle instance de EnvironmentForest est créée à partir du fichier et d'infos précédemment sauveés (dico), ainsi qu'à partir des paramètres de FenetreMain. L'instance courante de ForestEnvironmentView prend comme modèle cette dernière. Enfin, un réaffichage général est provoqué ("*self displayOn: self graphicsContext*").
 - **selectPlace: aPlace.** Cette méthode est appelée par la méthode **redButtonActivity** de ForestEnvironmentController (voir plus bas). Cette dernière récupère les coordonnées d'un point sur lequel on a cliqué, elle les fournit au modèle (instance de ForestEnvironment) qui lui renvoie l'identité du placeau les contenant. La méthode **selectPlace: aPlace** renseigne la variable *selectedPlace* ainsi qu'une instance de FenetrePlaceau précédemment créée à laquelle elle demande de changer d'emplacement et de se positionner sur le nouveau placeau.
- Donc, cette classe peut créer des instances des classes FenetrePlaceau (méthode **zoomSelectedPlace**) et ForestEnvironment (**loadNewEnvironmentForest**).

Variables d'instance.

- *arbresVisu* : variable booléenne initialisée à "true". Fonctionne avec *arbresFlag* de FenetreForet. Contrôle l'affichage de l'information correspondante.
- *chablisVisu* : variable booléenne initialisée à "false". Fonctionne avec *chablisFlag* de FenetreForet. Contrôle l'affichage de l'information correspondante.
- *echelleGraphique* : elle est fixée à 4@4 au moment de l'initialisation. C'est le nombre de pixels contenus dans 1 cm.
- *fenetreMere* : il s'agit de l'instance initialement appelante de FenetreForet. L'affectation est faite par cette dernière.
- *nbArbresVisu* : variable booléenne initialisée à "false". Fonctionne avec *nbArbresFlag* de FenetreForet. Contrôle l'affichage de l'information correspondante.
- *nbMortsVisu* : variable booléenne initialisée à "false". Fonctionne avec *nbMortsFlag* de FenetreForet. Contrôle l'affichage de l'information correspondante.

- *selectedPlace* : la méthode **selectPlace: aPlace** l'identifie à *Array with: aPlace*, sachant que *aPlace* est un plateau sur lequel l'utilisateur a cliqué (méthode **sendPlaceContaining: aPoint** de ForestEnvironment).
- *surfaceTerriereVisu* : variable booléenne initialisée à "false". Fonctionne avec *surfaceTerriereFlag* de FenetreForet. Contrôle l'affichage de l'information correspondante.
- *tailleFenetreY* : dimension verticale du monde représenté : "*tailleFenetreY:= (self model worldRealDimension*self echelleGraphique) y*".

2.2.2 - ForestEnvironmentController

Cette classe permet de redéfinir l'activité de la souris lorsqu'elle est positionnée sur la carte affichée dans FenetreForet (ie sur l'instance de ForestEnvironmentView).

4 méthodes correspondent aux 4 articles du menu déroulant. Elles vérifient d'abord qu'il existe un plateau sélectionné (ie la variable *selectedPlace* de ForestEnvironmentView n'est pas nulle). Elles permettent 1) d'inspecter les arbres (le premier arbre de la liste des arbres correspondant au plateau sélectionné), 2) d'inspecter l'instance de ForestEnvironment qui sert de modèle à la vue, 3) d'inspecter le plateau sélectionné, 4) d'ouvrir une fenêtre sur le plateau sélectionné (méthode **zoomPlaceau**).

La méthode **redButtonActivity** permet de redéfinir l'activité du bouton de gauche : elle récupère les coordonnées du "clic", demande à son modèle (ie ForestEnvironment) de lui identifier le plateau correspondant et transmet son identité à sa vue. Donc, résultat d'un "clic" : un plateau sélectionné (variable *selectedPlace* de la vue modifiée).

2.2.3 - GenericView

Cette classe donne naissance à PlaceauView. Ses méthodes permettent de multiplier les coordonnées d'un point par un facteur et de déterminer les dimensions d'une fenêtre.

Variables d'instance.

- *dimension* : dimension (point) d'une fenêtre qui vient d'être ouverte.
- *origin* : initialisée à 0@0.
- *factor* : initialisé à 10@10 (?). Redéfini dans PlaceauView : méthode **setFactor** qui l'identifie au rapport "*dimension / self placeau size*", avec *dimension* = taille de la fenêtre ouverte.

2.2.4 - PlaceauView

Hérite de GenericView. Une instance de cette classe est créée par FenetrePlaceau. Elle crée elle-même, au moment de son initialisation, une instance de PlaceauController : "*self controller: PlaceauController new initialize*".

Deux méthodes permettent, comme pour ForestEnvironmentView, de convertir les coordonnées réelles en coordonnées pixel en tenant compte de l'inversion de l'axe des y, et vice-versa.

Méthodes très importantes :

- **displayOn: aGraphicsContext**. Elle détermine la taille du rectangle correspondant à la fenêtre qui a été ouverte. Elle calcule "factor" (voir GenericView). Elle prend ensuite les arbres du plateau les uns après les autres, vérifie quelle couleur a été affectée à l'espèce correspondante (couleur contenue dans SelvaOptions – voir Options – et choisie par l'utilisateur à travers FenetreChoixCouleur et FenetreOptionsCouleurs – voir plus haut –), vérifie si l'arbre a été sélectionné pour une éclaircie auquel cas la couleur attribuée est la couleur de sélection (également répertoriée dans SelvaOptions) puis affiche les arbres. En final, la couleur de sélection est "armée" ("*self graphicsContext paint: (SelvaOptions couleurs at: #selection)*").
- **displayTree: aTree diametre: diam**. Affiche l'arbre comme un cercle de centre = aux coordonnées cartésiennes après conversion et de rayon proportionnel au diamètre (et fonction de la taille de la fenêtre et du placeau). **NB** : une méthode permet d'effacer un arbre en le réaffichant avec la couleur du fond !

- **changeSelection: old new: new.** Affiche "old" avec la couleur correspondant à son espèce, et "new" avec la couleur de sélection.

- **newSelection: new.** Méthode appelée notamment par FenetreEclaircie. Vérifie le nouvel état de l'arbre (#vivant si désélection, non #vivant si sélection pour éclaircie) et lui affecte soit sa couleur d'origine soit la couleur de sélection. Le plateau est ensuite averti du changement.

- **treeSelected: aPoint.** Cette méthode est appelée par PlaceauController lorsque l'on clique (avec le bouton de gauche) sur un point du plateau. Les coordonnées du point sont transmises à PlaceauView. La méthode demande alors au modèle (Placeau) de lui fournir l'identité de l'arbre présent sur ces coordonnées (après conversion) et elle envoie cette identité à la fenêtre mère (FenetrePlaceau) par l'intermédiaire de la méthode **arbreSelection: selection**. Cette dernière cherche l'arbre concerné dans *listeArbres* et, je suppose, place le curseur vidéo inversée sur la ligne correspondante de la liste (*listeNomsArbres*).

NB. L'instance appelante de FenetrePlaceau est affectée à la variable *fenetreMere* lorsqu'elle crée l'instance de PlaceauView.

Variables d'instance.

- *tailleFenetreY* : dimension verticale de la fenêtre ouverte ("*dimension y*").

2.2.5 - PlaceauController

Une instance de cette classe est créée par PlaceauView, qui la prend explicitement comme "controller" (voir précédemment).

Deux méthodes seulement :

- **controlActivity**, qui pilote le fonctionnement de la souris et du curseur ;

- **redButtonActivity: point**, qui redéfinit l'activité du bouton de gauche de la souris. En l'occurrence, celui-ci transmet à PlaceauView les coordonnées du point sur lequel on a cliqué.

2.2.6 - ArrayView

Cette classe sert de niveau hiérarchique supérieur aux classes EffectifsDiametreView (utilisée par FenetreDiametre), EvolutionView (utilisée par FenetreEvolution) et TreeView (utilisée par FenetreArbre) : voir ces classes plus bas.

Le but de la manoeuvre consiste à afficher une information (y) sous forme de barres, éventuellement de barres d'histogramme.

Méthodes très importantes :

- **displayOn: aGraphicsContext.** On commence par déterminer la taille de la fenêtre ouverte, puis on récupère les infos qui doivent être représentées en x et en y, on calcule un facteur d'affichage *factY* (méthode **findFactorY: yTab** qui recherche la plus grande valeur des y à représenter et s'arrange pour qu'elle occupe les 3/4 de la hauteur de la fenêtre : voir plus bas la liste des variables d'instance) et on affiche pour chaque x la valeur de y sous forme de barre : méthode **afficheBarres: aGraphicsContext xTab: xTab yTab: yTab**.

- **afficheBarres: aGraphicsContext xTab: xTab yTab: yTab.** On récupère un tableau de libellés qui seront affichés en haut de chaque barre. Pour chaque valeur contenue dans le tableau des y, on trace un rectangle de largeur "*pas*" (30 pixels) et de hauteur la valeur de y multipliée par factY. L'origine des x est fixée à 10 pixels en partant de la gauche, et celle des y à 10 pixels + 10 autres correspondant à la place laissée pour les "étiquettes". Chaque étiquette (placée sous le rectangle) correspond à la valeur x du couple (x, y) examiné. Les labels sont placés à 10 pixels au-dessus du rectangle. Ils doivent faire moins de 4 caractères et un test d'égalité entre 2 labels consécutifs est réalisé : s'ils sont identiques, on remplace le 2ème par "=" et cela 3 fois de suite avant de réafficher une valeur. On aura par exemple : 31.5 = = = 31.5 32.0, etc.

NB. Les méthodes **labelTab**, **tableauX** et **tableauY** sont redéfinies dans les classes des niveaux hiérarchiques inférieurs.

Variables d'instance.

- *dimension* : initialisée à 500@500, mais redéfinie par la suite comme la taille de la fenêtre ouverte ("*fenetre:= aGraphicsContext clippingBounds. dimension:= (fenetre width) @ (fenetre height)*").

- *factY* : calculé par la méthode **findFactorY: yTab**. Il vaut "*factY:= [3*(dimension y)/4] / max*", max étant la valeur maxi. contenue dans le tableau des y. Il est initialisé par défaut à 5.

- *indiceDeDepart* : initialisé à 1.

- *origin* : initialisée à 10@10.

- *pas* : initialisé à 30 (pixels).

2.2.7 - ArrayViewAvecMoyenne

Cette classe hérite de ArrayView. Elle permet d'afficher les mêmes infos mais en plus, elle calcule et met à jour en permanence la moyenne générale des y qu'elle affiche dans un coin (supérieur gauche : position 50@15, ne pas oublier que l'origine chez SmallTalk est située en haut à gauche) de la fenêtre. Elle sert de supérieur hiérarchique à EvolutionView.

La moyenne des y est calculée par la méthode **miseAJourMoyenne: yTab** selon la formule :

- soient $a_1, a_2, \dots, a_i, a_{i+1}$;

- $m_i = a_1 + a_2 + \dots + a_i$;

- $m_{i+1} = [(m_i * i) + a_{i+1}] / (i + 1)$.

La méthode **afficheMoyenne: aGraphicsContext yTab: yTab** gère l'ensemble calcul (appel à la méthode précédente) + affichage.

Variables d'instance.

- *cardinal* : initialisé à 0. Il s'agit du nombre de valeurs prises en compte pour le calcul de la moyenne.

- *moyenne* : résultat du calcul, affiché à chaque cycle de fonctionnement du simulateur.

2.2.8 - TreeView

Cette classe hérite de ArrayView. Deux instances sont créées par FenetreArbre qui leur attribue comme modèle une instance de Arbre. FenetreArbre passe à chacune des deux vues respectivement les étiquettes #voisins ou #diametre selon l'information y à traiter (méthode appelée : **visualiser: aspect**).

Toutes les infos à traiter sont stockées dans un tableau appartenant à Arbre et auquel on accède par une méthode **evolution**. Ce tableau a l'allure suivante :

Numéro du cycle	Diamètre	Nombre de voisins
...

Pour représenter les diamètres, **tableauY** récupère l'info. de la 2ème colonne et pour représenter le nombre de voisins "dominants", **tableauY** récupère la 3ème colonne. Dans tous les cas, **tableauX** stocke les nombres allant de 1 à "*self model evolution size*". Ces numéros sont convertis en chaînes de caractère au moment de l'affichage.

Variables d'instance.

- *visualiser* : contient l'étiquette #voisins ou #diametre. Lorsque FenetreArbre crée les instances de TreeView, les messages passés sont :

"*diametreView:= TreeView model: arbreModel. diametreView visualiser: #diametre.*"

voisinsView:= TreeView model : arbreModel. voisinsView visualiser: #voisins."

2.2.9 - EffectifsDiametreView

Cette classe hérite de ArrayView. 3 instances sont créées par FenetreDiametre, qui leur passe comme modèle l'instance de Evolution créée par ForestEnvironment. Chaque vue permet de visualiser un élément différent : une clé est passée par l'intermédiaire de la variable *elementAVoir*, qui fonctionne comme la variable *visualiser* de TreeView. Ex : "(view1:= EffectifsDiametreView model: *evolution*) *elementAVoir*: #*effectifParDiametre*".

tableauX et **tableauY** appellent des méthodes de Evolution pour récupérer les infos à afficher.

Variables d'instance.

- *cycle* : initialisé à 0. Il correspond au cycle pour lequel les infos doivent être affichées. Il est modifié par FenetreDiametre (méthode **cycleChange**) soit lorsque l'utilisateur appuie sur les flèches droite et gauche de la fenêtre, soit par incrémentation automatique pendant que la simulation se déroule (si le bouton "dernier cycle" est activé) : méthode **update: anAspect with: aParametre** de FenetreDiametre.

- *elementAVoir* : étiquette passée par FenetreDiametre. A l'initialisation, il s'agit de #*effectifParDiametre*, #*chablisParDiametre* et #*mortSurPiedParDiametre* respectivement pour les 3 vues créées. Lorsque l'utilisateur, ensuite, fait un choix dans les menus, les variables *aspectView1*, *aspectView2* et *aspectView3* de FenetreDiametre sont modifiées et leur valeur renseigne *elementAVoir*.

2.2.10 - EvolutionView

Cette classe hérite de ArrayViewAvecMoyenne. Une instance est créée par FenetreEvolution qui lui passe comme modèle l'instance de Evolution créée par ForestEnvironment.

Plusieurs infos peuvent être visualisées dans cette vue. L'utilisateur les choisit grâce au menu déroulant géré par FenetreEvolution (voir variable *menuEvolutionHolder*). Le choix est passé à EvolutionView par le message "**visuChanged: (menuEvolutionHolder value)**", qui renseigne la variable d'instance *visualiser*. Par défaut, l'info. affichée est #*nbArbres* (ie l'effectif par hectare d'arbres vivants présents sur la parcelle étudiée).

tableauY envoie au modèle (Evolution) un message destiné à récupérer l'info. à visualiser.

Variables d'instance.

- *visualiser* : voir texte précédent. Stocke "**menuEvolutionHolder value**", ie l'info. que l'utilisateur a choisie au niveau du menu de FenetreEvolution.

2.3 - Les modèles

2.3.1 - EnvironmentForest

Sous-classe de la classe Agent. Une instance de cette classe est créée par FenetreMain :

lancement: icone >> ... >> initEnvironmentFic >> "envt:= EnvironmentForest new". C'est le coeur de SELVA. EF gère toutes les activités de l'univers à travers sa méthode **act** (voir plus loin). Elle crée les placeaux, crée les arbres à travers la lecture des fichiers FIC, donne la main aux arbres pour qu'ils analysent leur environnement, décident de leur sort et agissent, donne la main aux placeaux pour qu'ils s'occupent du recrutement, etc. Elle s'occupe de la mise en forme des infos pour les sorties fichiers.

EF est responsable de la création des instances de différentes classes : Placeau (une instance pour chaque cellule de l'univers), EspeceGrise (une instance par arbre), Evolution (une instance par simulation), ForestKernelScheduler (qui crée un process actif), ForestRandomUnif (qui fournit des nombres aléatoires), ForestFile (qui permet de lire un fichier '.vivants' ou '.morts'), WriteFile (qui permet d'écrire des infos dans des fichiers ASCII).

La méthode **initEnvironmentFic: nomFichier** de FenetreMain (appelée par **initEnvironmentFic** après récupération du nom du fichier d'entrée) s'occupe de créer l'instance de EF, puis de mettre en place tous les éléments de l'univers à travers différentes méthodes :

- **initializeEnvironmentSelonFenetreMain**: **envt** (méthode de FenetreMain, que nous appellerons aussi FM) : passe à EnvironmentForest tous les paramètres définis au départ par l'utilisateur. A savoir : (a) la valeur de *stadeJuvenile* (booléen, *stadeJuvenileInactif* de FM) permettant d'activer le modèle complet de régénération ; (b) la valeur de *nomFichierIn* (*nomRepertoire* + *ficList* de FM) ; (c) la valeur booléenne de *autoStart* (*autoLancement* de FM) permettant d'envoyer la simulation sans affichage préalable de la carte dans FenetreForet ; (d) la valeur de *nbCyclesAvantOubli* (*cyclesSauve* de FM) ; (e) la valeur booléenne de *stopSiDemieVieAtteinte* (*demieVie* de FM) ; (f) la valeur de *cyclesAEffectuer* (*nbMaxCycles* de FM) ; (g) la valeur de *repertoireSortie* (variable du même nom de FM).

- **initialize**: **dim@dim** (méthode de EnvironmentForest) : *dim* est décodée par FenetreMain à la lecture du nom du fichier d'entrée. Il s'agit de la dimension du monde en nombre de placeaux (= paramètre "*cellsSize*", qui renseigne la variable d'instance *size* de EF). Première méthode d'initialisation, elle crée une liste ordonnée de tous les placeaux de l'univers (*cells*), des arbres vivants et des arbres morts, initialise certaines variables et crée les instances de Placeau (méthode **initializeCells**), de Evolution et de ForestKernelScheduler (méthode **initializeScheduler**).

- **initializeSysteme** (méthode de EnvironmentForest) : elle initialise les variables de classe de EspecesGrise, les variables de classe de Placeau et crée une instance de ForestRandomUnif.

A ce stade, les placeaux sont créés et leurs coordonnées absolues sont renseignées en tenant compte du fait que le monde est toroïdal. Il reste à "fabriquer" les accointances et à créer les arbres. Ces ordres sont donnés par FenetreForet à travers sa méthode **postOpenWith**: **aBuilder** (fin des instructions de la méthode **lancement**: **icone** de FenetreMain : "*fenetre openInterface: #windowSpec*" >> "*self postOpenWith: aBuilder*"). Les accointances sont créées par appel à la méthode **initAccointancesForToroidaleWorld**: (**envt class acc8**) de EF et les instances d'arbres à travers la méthode **lireForetInFichier**.

L'instruction "**initializeScheduler**" crée une instance de ForestKernelScheduler et la méthode **initializeProcess** de cette dernière est activée. Le process créé est le suivant :

```
process := [
    [true] whileTrue: [
        self environmentForest act.
        self incAbsoluteCounter.
        Processor yield.
    ]
] newProcess.
```

2.3.1.1 - La méthode **act**

Fondamentale : elle règle toute l'activité au cours d'un cycle.

On commence par tester l'arrivée en fin de simulation ("**self finDeSimulation ifTrue:**") : vrai si le temps de demie-vie est atteint, ou si la variable *cyclesRestantAEffectuer* est égale à 0. Dans ce cas, une sortie automatique des fichiers est réalisée (affectation de noms par défaut) et l'instance de EnvironmentForest avertit ses dépendances que la simulation est terminée : "**self changed: #terminate**".

Dans le cas contraire, on commence par incrémenter *cycleCourant* (initialisé à 0) et décrémenter *cyclesRestantAEffectuer* (initialisé à *nbMaxCycles* dans FenetreMain). La méthode **initCycle** met à 0 les variables *nbMortsInCycle* et *nbNaissancesInCycle* et à -1 *surfaceTerrierePondereeMaxi*. La méthode **initialisePlaceauxPourCycle** envoie à toutes les instances de Placeau le message **initCycle**, qui réalise un calcul de la surface terrière pondérée puis une recherche de la valeur maximum afin de déterminer *surfaceTerrierePondereeMaxi*. La valeur de l'agent courant est mise à 0, puis les actions s'enchaînent de la manière suivante : (a) les arbres analysent leur environnement (à l'aide des infos fournies par les placeaux) et décident de leur devenir (méthode **deliberationArbres**) ; (b) les arbres candidats à la reproduction disséminent leurs graines (méthode **effectationArbresDisseminateurs**) ; (c) les placeaux examinent les possibilités de recrutement (méthode **deliberationPlaceaux**) ; (d) les arbres agissent (méthode **effectationArbres**) ; (e) les placeaux réalisent le recrutement (méthode **effectationPlaceaux**).

Plus en détail :

- **deliberationArbres** prend les arbres (contenus dans *listeArbres*) les uns après les autres, vérifie qu'ils ont bien l'étiquette #vivant (variable *etat*) et leur demande : **choisiTonDestin** (méthode de EspeceGrise, voir plus bas). Le but du jeu consiste à renseigner la variable *destin* de chaque arbre (voir plus bas le chapitre concernant EspeceGrise). **NB.** L'appel au module Detection se fait en cours de délibération.

Avant que la main soit donnée aux arbres de la liste, on crée *listeArbresDisseminateurs*. En effet, si l'arbre s'est connecté au module DeliberationAngelique, il a la possibilité de devenir reproducteur auquel cas il avertit EnvironmentForest qui l'ajoute à cette liste : **arbre choisiTonDestin >> moduleDeliberation doAct: self >> self fleuri: moduleAgent >> moduleAgent environnement arbreDisseminateur: moduleAgent >> listeArbresDisseminateurs add: moduleAgent**. **NB.** Avant d'être un disséminateur, l'arbre doit avoir été déclaré "en croissance" : le premier champ de sa variable *destin* (qui en comporte alors 4) doit contenir l'étiquette #croissance.

- **effectioArbresDisseminateurs** prend les arbres de la liste *listeArbresDisseminateurs* et leur demande : **dissemineGraines**, avant de remettre la liste à 0. Cette fonction, par appel au module Effectio, a pour effet de remplir les stocks de graines des placeaux proches et lointains.

- **deliberationPlaceaux** prend chaque placeau de la liste *cells* et lui demande : **choisiTonDestin: stadeJuvenile** (*stadeJuvenile* = booléen). Voir § 2.3.2. La mise en oeuvre de cette méthode aboutit au "remplissage" d'une collection *arbresANaitre* (variable de Placeau) constituée de couples (code espèce, nombre de tiges à recruter). Si *stadeJuvenile* est à *false*, un seul couple est présent, celui de l'espèce grise, et le modèle de recrutement est directement mis en oeuvre par Placeau. Si *stadeJuvenile* est à *true*, un appel aux classes concernant les juvéniles est réalisé, en commençant par StadeJuvenile.

- **effectioArbres** prend les arbres de *listeArbres* un par un et leur demande : **"a effectTonDestin"**. L'arbre passe alors à Effectio(Grise) le tableau contenant son destin, et l'instance de cette classe s'occupe de la mise à jour de toutes les variables concernées, et demande aux placeaux et à l'environnement d'en faire autant (voir plus bas les chapitres EspeceGrise et EffectioGrise).

- **effectioPlaceaux** prend les placeaux les uns après les autres et leur demande **"c effectTonDestin"**. Le placeau examine alors sa liste *arbresANaitre*, et procède aux recrutements indiqués (méthode **recrutement: cycleCourant codeEspece: code**, voir § 2.3.2). **ATTENTION** : les coordonnées affectées aux nouveaux arbres le sont par rapport au placeau concerné. Comme l'origine respective des coordonnées des arbres et des placeaux est différente, le placeau redonne la main à EnvironmentForest pour qu'il réaffecte un placeau propriétaire ("**self owner addTree: arbre voisins: true**") : à partir de ce moment, les procédures mises en oeuvre sont les mêmes que pour un arbre lu initialement dans un fichier '.vivants'.

A la fin de ces instructions, la liste *arbresMorts* de EF se trouve remplie. EF les prend un à un et les élimine de sa liste d'arbres vivants (*listeArbres*).

Une instruction **initializeCycle: self** envoyée à Evolution permet de mettre à jour son dico. *tabMemoire*, puis l'instance d'EnvironmentForest indique à ses dépendances qu'un nouveau cycle va démarrer : "**self changed: #nouveauCycle**".

Enfin, un "nettoyage" des arbres morts a lieu au travers de la méthode **suppressionDesMorts** : EF prend les arbres de sa liste *arbresMorts* et vérifie que leur date de mort n'est pas inférieure à la valeur "**self cycleCourant - nbCyclesAvantOubli**". Si c'est le cas, l'arbre mort examiné avertit son placeau propriétaire qu'il doit le supprimer de sa propre liste *arbresMorts*, et EnvironmentForest en fait autant.

2.3.1.2 - Les sorties de fichiers

Lorsque la fin de la simulation arrive (test réalisé en début de méthode **act**), une sortie automatique des fichiers est enclenchée (avec affectation de noms de fichiers par défaut). A noter qu'une sortie peut également être requise manuellement à travers le menu déroulant de FenetreForet : l'item "Sauvegarde fichiers" envoie à EnvironmentForest l'ordre : "**sortiesFichiers: true**", la valeur *true* signifiant qu'il sera demandé à l'utilisateur d'entrer manuellement le nom des fichiers de sauvegarde.

La méthode **sortiesFichiers** : **booléen** gère les sorties. Plusieurs types de fichiers peuvent être fabriqués. Ils correspondent à 5 variables de **FenetreMain**, que l'utilisateur met à *vrai* ou *faux* en début de travail : *fichierFicSimple*, *fichierFicComplet*, *fichierEvolution*, *fichierForest* et *fichierStadeJuvenile*.

- *fichierFicSimple* et *fichierFicComplet*, s'ils sont à *vrai*, entraînent l'utilisation de la même méthode : **sortieFichierFic: confirmation complet: booléen**. Le booléen vaut *false* dans le premier cas, *true* dans le deuxième. Le nom des fichiers par défaut est fabriqué à partir de la variable *repertoireSortie* à laquelle on ajoute diverses infos : la date, le n° de la simulation, suivis de *fic_simul_simple* ou *fic_simul_complet* selon le cas. On détermine ensuite le cycle à partir duquel il faut écrire les infos : "**debut:= cycleCourant - nbCyclesAvantOubli + 1**" (la variable *cycleCourant* récupère la valeur de la variable *cyclesSauve* de **FenetreMain** à l'initialisation). Puis une instance de **WriteFile** est créée (cette classe facilite l'écriture par la gestion du pointeur). Résultat :

3 lignes explicatives sont inscrites en début de fichier :

SIMULATION <date>, <heure>, Parcelle n° : <noParcelle>

Fichier d'entrée : <nomFichierIn>, Cycle début : <debut>, Cycle fin : <cycleCourant>. Densité de sauvegarde : tous les <densiteDeSauvegarde> cycles.

LEGENDE : n° carré ; identifiant arbre ; code espèce ; position x (dm) ; position y (dm); liste < diamètres (-1 -> inconnu 999... -> mort).

Si on a demandé la sortie d'un fichier FIC complet, la 3ème ligne est la suivante :

LEGENDE : n° carré ; identifiant arbre ; code espèce ; position x (dm) ; position y (dm); liste < diamètres (-1 -> inconnu 999... -> mort) ; voisins dominants (-1 : inconnu) ; surface terrière ; ATTENTION !!!!! Les IC correspondent aux valeurs utilisées pour le calcul du diamètre qui les précède !

Ensuite, **EF** parcourt la liste de ses arbres vivants, puis celle de ses arbres morts en faisant appel à la méthode de **EspecGrise** **sortieFichierFic: debut fin: cycleCourant complet: complet**. Cette dernière renvoie les infos demandées en consultant les variables d'instance, notamment le tableau *evolution* qui stocke, à chaque cycle, les infos diamètre, NBD et surface terrière des voisins (voir **EspecGrise**, plus bas).

- *fichierEvolution*, s'il est à *vrai*, entraîne l'utilisation de la méthode **saveEvolution: confirmation**. Le nom du fichier par défaut est fabriqué à partir de la variable *repertoireSortie* à laquelle on ajoute diverses infos : la date, le n° de la simulation (*noSimulation*) et '.evolution'. Il s'agit d'un fichier BOSS : on crée donc une instance de **BinaryObjectStorage** et, après avoir "mis de côté" ses dépendances (pour les récupérer ensuite), on sauve directement l'instance de **Evolution**. Celle-ci pourra être récupérée par une opération symétrique pilotée par la méthode **load: fileName** de **Evolution**.

- *fichierForest*, s'il est à *vrai*, entraîne l'utilisation de la méthode **saveEnvironmentForest: confirmation**. Le nom du fichier par défaut est cette fois fabriqué à partir de *nomFichierIn* auquel on ajoute l'extension '.forest'. Il s'agit d'un fichier BOSS : la démarche est exactement la même que pour **Evolution**, sauf que le nombre d'éléments mis de côté (méthode **saveForBoss**) puis récupérés (méthode **retraveForBoss**) est plus important. **NB** : la sauvegarde se fait simplement par "**nextPut: self.**" ! La méthode **saveForBoss** remplit un dictionnaire avec les objets qui ne peuvent être sauvés : les dépendances de **EF**, le sablier (qui stocke l'instance, me semble-t-il, de **FenetreMain**), la variable *currantAgent*, l'instance de **ForestKernelScheduler**, et les met à *nil*. La méthode **retraveForBoss** réalise l'opération inverse.

- *fichierStadeJuvenile* s'il est à *vrai*, entraîne l'utilisation de la méthode **sortieFichierStadeJuvenile: confirmation**. Si la variable *stadeJuvenile* est à *false*, rien ne se produit. Dans le cas contraire, le nom du fichier par défaut est fabriqué à partir de la variable *repertoireSortie* à laquelle on ajoute diverses infos : la date, le n° de la simulation (*noSimulation*) et l'extension '.juvenile'. Le "remplissage" se fait de manière un peu complexe : il faut d'abord récupérer, chez le premier plateau de la liste *cells*, la liste des libellés des espèces (clés récupérées dans le tableau *evolution* de **Placeau**) concernées par le modèle complet de régénération. Puis, pour chacune de ces espèces, on crée une instance de **WriteFile** : on fabrique donc un fichier par espèce concernée !!!.

Le nom du fichier est complété par '_*nom de l'espèce*>' et 3 lignes d'en-tête sont écrites :

SIMULATION <date>, <heure>, Parcelle n° : <noParcelle>. ESPECE : <libelleEspec>.

Fichier d'entrée : `<nomFichierIn>`, Cycle début : `<debut>`, Cycle fin : `<cycleCourant>`. Densité de sauvegarde : tous les `<densiteDeSauvegarde>` cycles.

LEGENDE : coordonnées placeau ; `<surface terre pondérée ; graines; plantules ; gaules ; perches ; recrutés >`.

2.3.1.3 - Le point sur : la création et l'initialisation des placeaux

Chronologiquement, les ordres émis sont les suivants :

- "`cells := (OrderedCollection new: cellsSize x * cellsSize y)`", avec `cellsSize := dim@dim`, paramètre passé par `FenetreMain` (méthode `initialize:<>`, voir plus haut).

- "`self initializeCells`" (méthode `initialize:<>`) : crée et ajoute à la liste `cells` autant d'instances de `Placeau` qu'il en faut dans l'univers (ie `dim*dim`). **NB**: cela suppose que la dimension de l'univers est au départ donnée en nombre de placeaux constituants (c'est le cas, dans l'en-tête des fichiers de Dario), ce qui n'est pas logique. A REPRENDRE EVENTUELLEMENT. Par ailleurs, des coordonnées sont affectées à toutes les instances de la manière suivante : si l'univers fait 1 ha, sa dimension est de 10@10, 100 instances de `Placeau` sont créées et "numérotées" de 1 à 100 dans `cells` et les coordonnées sont : 0@0, 0@1, 0@2, ..., 0@9, 1@0, 1@1, ..., 1@9, etc. Le premier placeau a pour coordonnées 0@0 (l'origine est située **en haut à gauche de l'écran** !!!), le 10ème 0@9, le 11ème 1@0, le 20ème 1@9, le 100ème 9@9 (remplissage d'une grille de gauche à droite et de haut en bas).

- "`Placeau classPlaceauSize: taillePlaceau`" (méthode `initializeSysteme`) : renseigne la variable `Size` de `Placeau`, c'est-à-dire fixe les dimensions (en mètres) de ses instances.

- "`Placeau initClassVariables: worldRealDimension`" (méthode `initializeSysteme`) : renseigne la variable `WorldDim` de `Placeau` (pour un univers de un hectare, `worldRealDimension` vaut 100@100) et initialise la représentation géométrique d'un placeau.

- "`envt initAccointancesForToroidaleWorld: (envt class acc8)`" (méthode `postOpenWith: aBuilder` de `FenetreForet`) : `acc8` est une méthode de la classe `EF` qui renvoie une liste ordonnée de 8 points correspondant aux placeaux proches d'un placeau pris comme origine :

5 (SO) -1@-1	2 (Sud) 0@-1	6 (SE) 1@-1
1 (Ouest) -1@0	0 (Centre) 0@0	3 (Est) 1@0
8 (NO) -1@1	4 (Nord) 0@1	7 (NE) 1@1

La liste renvoyée est `{(-1@0), (0@-1), (1@0), (0@1), (-1@-1), (1@-1), (1@1), (-1@1)}`. **NB** : comme `ForestEnvironmentView` inverse l'image à l'écran, c'est bien le placeau Sud qui se trouve en position 2 et le placeau Nord qui se trouve en position 4.

La méthode `initAccointancesForToroidaleWorld`: liste récupère chaque élément de `cells`, et lui associe la liste des 8 points référencés cette fois par rapport à la position exacte du placeau examiné. Puis elle traite le cas de toutes les bordures de l'univers : Ouest et Sud, Nord et Est, ainsi que le cas des 4 points des "coins". Les voisins proches situés en dehors des limites de cet univers voient leurs coordonnées modifiées pour se retrouver "de l'autre côté" de la parcelle et ainsi boucler un tore. Comme les points modifiés sont d'abord extraits de la liste initiale, avant d'y être rajoutés, l'ordre qui correspondait à une coïncidence avec les points cardinaux est perdu : on refait donc passer la liste finale par la méthode `makeMyAccointance` de `Placeau`, de manière à récupérer une affectation définitive correcte (ainsi que l'identité des placeaux correspondant aux différents points) : voir plus bas.

2.3.1.4 - Le point sur : la création et l'initialisation des arbres.

Chronologiquement, les ordres émis sont les suivants :

- *EspeceGrise initializeClassVariables* (méthode **initializeSysteme**). Cette méthode de classe initialise la variable de classe *AnneesParCycle* à 3 et crée une instance de toutes les sous-classes de la classe ModuleInstanceUnique. Par la suite, chaque arbre nouvellement créé se rattachera à ces différentes instances.

- *envt lireForetInFichier* (méthode **postOpenWith: aBuilder** de FenetreForet) : on vérifie que (a) il n'existe pas d'instance mère de FenetreForet ; ou (b) si elle existe, l'option du menu correspondant à la lecture d'un fichier BOSS n'est pas activée ; ou (c) il n'existe pas de fichiers de type '.forest' (fichiers BOSS). Dans tous ces cas, la fonction **lireForetInFichierFic** est activée (sinon, le message **loadNewEnvironmentForest: (nomFichierIn, '.forest')** est envoyé à ForestEnvironmentView.

Après initialisation de l'instance d'Evolution ("*evolution initializeAnnee: noCampagne*", voir § 2.3.3), une première instance de ForestFile est créée pour décoder le fichier '.vivants'. A chaque lecture d'une ligne valide, l'instance d'Evolution est renseignée ("*evolution initializeParFichier: arrayTree*"), une nouvelle instance de EspeceGrise est créée et initialisée à partir des infos du fichier puis, à travers la méthode **addTree: <> voisins: <>**, elle est ajoutée à la liste des arbres vivants de l'univers (*arbresVivants*, ou *listeArbres*), à la liste des arbres vivants du plateau propriétaire après identification de celui-ci, et certaines variables sont incrémentées : *nbArbres*, *nbNaissancesInCycle*, *surfaceTerriere*.

Une fois lues toutes les lignes du fichier '.vivants', les arbres sont repris les uns après les autres et on leur associe une liste des voisins "lointains" (de toute taille), c'est-à-dire non situés dans les 8 placeaux de l'acointance : "*agent initializeAvecEnvironment: self listeArbres*" (méthode de EspeceGrise).

Enfin, une deuxième instance de ForestFile est créée pour lire le fichier '.morts' et achever ainsi d'initialiser l'instance d'Evolution.

NB. C'est la méthode **sendPlaceContaining: (unArbre position)** qui identifie le plateau propriétaire d'un arbre. Ce plateau n'est le bon QU'APRES INVERSION A L'ECRAN. Exemple : soit le point de coordonnées (31@21) : il s'agit de mètres et l'origine est située en bas à gauche de la parcelle. Le plateau propriétaire aura pour coordonnées 3@2 (placeau n°24 de la liste *cells*, au lieu du placeau n°74 auquel il appartient réellement) dans un repère dont l'origine se trouve en haut à gauche de la parcelle : il n'y a pas superposition.

2.3.1.5 - Les calculs toroïdaux.

Une série de fonctions sont définies pour pouvoir gérer le problème du tracé entre un chablis primaire et les chablis secondaires qu'il provoque, ce dans un monde toroïdal. En effet, le problème est qu'aucun point n'est unique : chaque placeau possède aux environs immédiats de l'univers 3 images équivalentes. Supposons un univers à 4 placeaux :

1	2	3 (a)	4	1	2	3 (a1)	4
5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	12
13	14	15 (b)	16	13	14	15	16
1	2	3 (a2)	4	1	2	3 (a3)	4
5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	12
13	14	15	16	13	14	15	16

Etant donné un chablis secondaire (b), il faut trouver le chablis primaire qui a causé sa mort (a). Connaissant les coordonnées de (a) et (b) dans l'univers d'origine, on cherche le point (a_i) le plus proche de (b) dans un monde toroïdal. On commence donc par rechercher toutes les "images (a_i) de (a) à l'extérieur de l'univers originel, puis on calcule les distances (non toroïdales) entre (b) et ces différents points. Le point retenu est le plus proche, et la droite tracée ira du point (b) à ce point.

Il faut ensuite procéder de la même manière pour le point (b) : chercher le point (b_i) le plus proche de (a) pour tracer une droite allant de (b_i) à (a). C'est à ce prix que le tracé semblera aller d'un bord à l'autre lorsque (a) et (b) sont sur les bordures de l'univers. Les méthodes concernées sont (a) **imagesDuPoint: p** ; (b) **imageNonToroidaleDe: b parRapportA: a** ; (c) **distanceNonToroidaleSqrt: p1 to: p2** et (d) **displayLineBegin: <> end: <> color: <>** de ForestEnvironmentView.

2.3.1.6 - Variables d'instance.

- **autoStart** : booléen qui reprend la valeur de **autoLancement** (variable de FenetreMain) lors de l'appel à la méthode **initializeEnvironmentSelonFenetreMain: envt** (voir en début de chapitre).

- **arbresMorts** : liste des arbres morts à l'issue d'un cycle. Cette liste est "nettoyée" dès que le temps de conservation des morts d'un cycle (indiqué par **nbCyclesAvantOubli**) est dépassé (méthode **suppressionDesMorts**).

- **arbresVivants** : liste des arbres vivants (méthode d'accès = **listeArbres**) ! Elle est alimentée par la méthode **addTree: unArbre voisins: boolean** (à la lecture initiale du fichier '.vivants' et, par la suite, au moment de **effectuationPlaceaux**, c'est-à-dire du recrutement). A la fin de chaque cycle, elle est mise à jour par élimination des arbres qui ont "atterri" dans **arbresMorts**.

- **cells** : liste des placeaux de l'univers, créé par la méthode **initialize: cellsSize**.

- **currentAgent** : cette variable est un "value holder" (méthode d'accès : **agentCourant**). Elle est initialisée à 0 par la méthode **act** à chaque nouveau cycle, et incrémentée (méthode **incAgentCourant**) par pas de 20 lors du passage en revue des arbres dans **deliberationArbres**, puis par pas de 10 dans **effectuationArbresDisseminateurs**, par pas de 10 lors du passage en revue des placeaux dans **deliberationPlaceaux**, par pas de 10 lors du passage en revue des arbres dans **effectuationArbres**, et enfin par pas de 10 dans **effectuationPlaceaux**. Elle est connectée à la variable **agentNumber** de FenetreForet, et sa valeur s'affiche en face de l'étiquette "Calcul".

- **cyclesAEffectuer** : initialisée par FenetreMain à partir de la valeur de sa variable **nbMaxCycles** et de la méthode d'accès **cyclesAEffectuer** (qui renseigne en même temps la variable suivante).

- *cycleCourant* : initialisée à 0 et incrémentée à chaque appel de **act**, avant que la main soit donnée aux agents !
- *evolution* : instance de Evolution, créée à l'initialisation et mise à jour dans **act**.
- *facteurAffichageNbArbres* : même commentaire que pour *facteurAffichageSurfaceTerriere*. La valeur maxi. déterminée est *maxNbArbres*, à partir de la variable *nbArbres* de Placeau.
- *facteurAffichageNbMorts* : même commentaire que pour *facteurAffichageSurfaceTerriere*. La valeur maxi. déterminée est *maxNbMorts*, à partir de la variable *nbArbresMorts* de Placeau.
- *facteurAffichageSurfaceTerriere* : variable mise à jour par la méthode **initializeFacteursAffichage** appelée par ForestEnvironmentView (méthode **displayOn: aGraphicsContext**). Cette méthode détermine la valeur maxi. (*maxSurfaceTerriere*) de la variable *surfaceTerriere* des placeaux de l'univers (méthode **maxParametres**). Elle calcule ensuite le facteur d'affichage de manière à ce que cette valeur maxi. soit représentée dans ForestEnvironmentView par un cercle de rayon (largeur du placeau / 2) !
- *listeArbresDisseminateurs* : liste renseignée par Deliberation(Espece). Un arbre qui a préalablement été déclaré en #croissance peut devenir reproducteur, et alimente alors cette liste (voir fonctionnement détaillé plus haut, dans la méthode **act**).
- *maxNbArbres* : valeur maxi de la variable *nbArbres* de Placeau.
- *maxNnMorts* : valeur maxi de la variable *nbArbresMorts* de Placeau.
- *maxSurfaceTerriere* : valeur maxi de la variable *surfaceTerriere* de Placeau.
- *nbArbres* : nombre total d'arbres vivants.
- *nbArbresAuDepart* : paramètre passé par FenetreMain (**initEnvironmentFic: <>** et **initEnvironmentBoss**). Il vaut *nbArbres* si *foretPartielle* (variables de FenetreMain) est vraie et 0 dans le cas contraire. **NB.** Ce paramètre intervient dans la lecture des fichiers '.vivants' et '.morts'. Lorsqu'il est différent de 0, il stoppe la lecture quand le nombre fixé d'arbres est atteint : comme les vivants sont chargés en premier, il ne peut pas y avoir d'arbres morts au départ dans une forêt partielle.
- *nbCyclesAvantOubli* : prend la valeur de *cyclesSauve* (variable de FenetreMain) lors de l'appel à la méthode **initializeEnvironmentSelonFenetreMain: envt** (voir en début de chapitre).
- *nbMortsInCycle* : variable mise à 0 par **initializeCycle** à chaque appel de **act**. Stocke le nombre de morts au cours du cycle.
- *nbNaissancesInCycle* : variable mise à 0 par **initializeCycle** à chaque appel de **act**. Stocke le nombre de recrutés au cours du cycle.
- *noCampagne* : paramètre passé par FenetreMain lors de **initEnvironmentFic: <>**. Il indique le n° de la campagne qui doit servir de cycle 0 (= démarrage de la simulation).
- *nomFichierIn* : nom générique des fichiers '.vivants' et '.morts'. Paramètre passé par FenetreMain, qui fabrique ce nom de la manière suivante : "**^(nomRepertoire value),',(ficListe selection)**". La méthode de EF appelée est **fichier: <> nbArbres: <>**.
- *repertoireSortie* : prend la valeur de la variable *repertoireSortie* (variable de FenetreMain) lors de l'appel à la méthode **initializeEnvironmentSelonFenetreMain: envt** (voir en début de chapitre).
- *sablier* : contient l'instance de FenetreMain. Elle est renseignée au moment de l'appel aux fonctions de FM : **initEnvironmentFic: <>** et **initEnvironmentBoss**.
- *scheduler* : renseignée par la méthode **initializeScheduler**. Elle contient l'instance de ForestKernelScheduler.
- *size* : il s'agit de la taille de l'univers en nombre de placeaux. Ce paramètre est passé par FenetreMain au moment de l'initialisation de EnvironmentForest : il est lu dans l'en-tête du fichier '.vivants'. *size* vaut 10@10 dans le cas d'une parcelle de 1 ha.
- *sortieFichierDemande* : cette variable est initialisée à *false*. Elle contrôle le déclenchement d'une sortie automatique des fichiers en fin de cycle (voir fin de la méthode **act**).
- *stadeJuvenile* : booléen qui reprend la valeur de *stadeJuvenileInactif* (variable de FenetreMain) lors de l'appel à la méthode **initializeEnvironmentSelonFenetreMain: envt** (voir en début de chapitre).

- *stopSiDemieVieAtteinte* : booléen qui reprend la valeur de *demieVie* (variable de FenetreMain) lors de l'appel à la méthode **initializeEnvironmentSelonFenetreMain**: **envt** (voir en début de chapitre).

- *surfaceTerriere* : ST totale de la forêt. Variable mise à jour à chaque appel de **addTree**: **unArbre voisins: boolean**, ainsi qu'au moment de l'ordre "*self effectioArbres*". Dans ce dernier cas, lorsqu'un arbre a pour destin #croissance, Effectio(Grise) modifie son diamètre et demande à Placeau de modifier sa variable *surfaceTerriere* en conséquence. A son tour, Placeau avertit EnvironmentForest (méthode **incSurfaceTerriere: delta**) qui fait de même.

- *surfaceTerrierePondereeMaxi* : initialisée à -1 en début de chaque cycle (**initializeCycle**). Lorsque la méthode **initializeCycle** de Placeau est appelée (par **initializePlaceauxPourCycle**, dans **act**), le calcul de *surfaceTerrierePonderee* est réalisé et Placeau transmet cette valeur à EnvironmentForest en appelant sa méthode **surfaceTerrierePondereeMaxiCandidate**: **<>**. C'est cette méthode qui renseigne la variable de EF. Elle sera utilisée par la suite dans les modules concernant le modèle complet de régénération.

- *tempsDesMorts* : initialisée à 1. C'est le nombre de cycles pendant lesquels un arbre mort peut être affiché à l'écran. Utilisée par la méthode **display: quoi on: aVue** de Placeau et **dateDeSouvenirDesMorts** de EF.

2.3.2 - Placeau

Sous-classe de la classe Agent. Les instances de cette classe sont créées par EnvironmentForest au moment de son initialisation : le message **initializeCells** constitue une liste de placeaux par appel à : "*Placeau new initialize: self*".

Placeau gère un certain nombre d'informations, en particulier il détient la liste des arbres vivants et des arbres morts et c'est lui qui décide du recrutement des arbres de toutes espèces (grise ou non) (méthode **nombreNaissancesArbresGris**). A ce titre, il crée une instance de EspeceGrise au moment du recrutement. Il utilise une instance de RandomUnif pour le tirage de nombres aléatoires concernant : 1) la détermination du nombre de recrutés "Gris" dans un cycle et 2) pour l'attribution de coordonnées à ces recrutés, à l'intérieur du placeau. ATTENTION : cette instance est créée par EnvironmentForest (**initializeSysteme**).

Placeau gère tous les stades de la régénération : #graines, #plantules, #gaules, #perches, #recrutes à l'aide du dico. *evolution*. (Voir ces stades au § 2.5).

Placeau fait appel à SelvaOptions pour récupérer certaines informations : valeur de #fondUni (méthode **afficheFond: ...**), initialisation du dico. *stadeJuvenile* (méthode **initialize: environmentForest**).

Par ailleurs, Placeau gère l'identification de ses voisins des 1ère, 2ème et 3ème couronnes dans des directions données. **NB** : le code est à reprendre car le monde est à l'envers !

		17	18	19		
	1	2	3	4	5	
28	16	5	2	6	6	20
27	15	1	0	3	7	21
26	14	8	4	7	8	22
	13	12	11	10	9	
		25	24	23		

0 : placeau sujet.

1 2 3 4 5 6 7 8 : placeaux de la 1ère couronne (stockés dans "*accointance*" = "*placeauxProches*").

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 : placeaux des 2ème et 3ème couronnes, stockés en début de la liste "placeauxLointains". A leur suite se trouvent les 12 placeaux des coins de la 3ème couronne.

Méthodes très importantes :

- **makeMyAccointance: aCollection.** *aCollection* est une série de 8 points (unité = nombre de placeaux) correspondant aux placeaux de la première couronne, ie aux "placeaux proches" du placeau sujet. Il s'agit d'ordonner les points puis d'identifier les placeaux concernés, et de les stocker dans la liste "accointance" appelée aussi "placeauxProches" à une place bien précise ! Cette place a été définie par Caruso de la manière suivante (voir schéma ci-dessus) : (1) = OUEST, (2) = NORD, (3) = EST, (4) = SUD, (5) = NW, (6) = NE, (7) = SE, (8) = SW. Elle est fonction de la valeur des coordonnées des points transmis dans *aCollection*. **MAIS** : il y a une erreur dans les tests sur les coordonnées : EST et OUEST sont inversés.

- **calculPlaceauxLointains.** Cette méthode stocke dans une liste (*placeauxLointains*) les 16 placeaux de la 2ème couronne (selon l'ordre mentionné sur le schéma), à l'aide de la méthode **contour: dim**, puis les 24 placeaux de la 3ème couronne à l'aide de la méthode **contourOrdonne**, en plaçant en fin de liste les 12 placeaux des coins, qui n'interviennent pas dans la compétition.

- **placeauxEnDirection: dir.** Cette méthode renvoie les deux placeaux de la 1ère et de la 2ème couronnes situés dans la direction "dir" spécifiée.

- **recursiveAccointance: tab ind: ind.** *tab* est une liste de directions et *ind* est un indice de démarrage sur la liste. La méthode permet de se déplacer à partir du placeau sujet vers l'un de ses placeaux proches ou lointains.

- **distanceBetween: p1 to: p2.** Cette méthode calcule la distance la plus courte entre 2 points, compte-tenu du fait que l'on travaille dans un monde toroïdal.

- **treeOnPosition: point.** Méthode appelée par PlaceauView, qui fournit à Placeau les coordonnées cartésiennes d'un point (en mètres) et demande à ce qu'on lui renvoie l'identité de l'arbre situé à proximité de ce point. Pour cela, la méthode recherche dans *listeArbres* tous les arbres situés à une distance du point inférieure à 2 fois leur diamètre, et elle renvoie l'identité du plus petit.

- **choisiTonDestin: executerStadeJuvenile.** Une première partie de la méthode concerne le recrutement des "arbres gris", ie sans distinction d'espèce et de plus de 10 cm de diamètre. Un appel à la méthode **nombreNaissancesArbresGris** permet de récupérer un nombre de tiges à recruter (modèle de recrutement : semi-stochastique, fonction de la ST cumulée sur le placeau concerné et sur ses voisins proches).

Une deuxième partie teste la valeur du paramètre booléen *executerStadeJuvenile* (c'est la variable *stadeJuvenile* de EnvironmentForest qui est passée) : s'il est faux, on sort de la fonction. Sinon, on initialise un nouveau "sous-dictionnaire" dans le dictionnaire *evolution* et on y renseigne, pour le cycle courant, la variable *surfaceTerrierePonderee*. Ensuite, les espèces du dictionnaire *stadeJuvenile* sont prises une à une et un appel à la classe StadeJuvenile ("*uneEspece calculRecrutesEtMiseAJour: self dicoEvolution: dicoCycle*") permet (a) de mettre à jour le sous-dictionnaire passé en paramètre (remplissage des champs #graines, #plantules, #gaules, #perches, #recrutes) et (b) de renvoyer au placeau le nombre de tiges à recruter au-dessus de 10 cm de diamètre.

Dans tous les cas, on renseigne la collection ordonnée *arbresANaitre* avec des couples (code espèce, nombre de tiges à recruter).

- **recrutement: cycleCourant codeEspece: code.** Méthode appelée pour chaque individu indiqué dans la collection *arbresANaitre*. Elle crée une nouvelle instance de EspeceGrise, lui affecte une position (x@y) au hasard, une date de naissance (= *cycleCourant*), lui demande de s'initialiser et renseigne EnvironmentForest sur son existence ("*self owner addTree: arbre voisins: true*"). Voir cette méthode dans le § 2.3.1.4. **NB** : cette méthode redemande à EF d'identifier le placeau d'appartenance de l'arbre : c'est utile, car le placeau propriétaire va changer (toujours pb. de l'inversion des origines) !). EF donne alors l'ordre à ce nouveau placeau de mettre ses variables à jour.

- **sortieFichierStadeJuvenile: libelleEspece debut: debut fin: fin.** Cette méthode renvoie à EnvironmentForest une chaîne de caractères constituée de : 1) les coordonnées du placeau (0@0 à 24@24), sachant que l'origine se situe en haut à gauche de l'écran, puis du cycle n° "debut" au cycle n° "fin", avec un pas spécifié par l'utilisateur au niveau de FenetreMain (voir variable *densiteSauvegarde*), les infos : 2) surface terrière

pondérée et 3) le nombre de graines, de plantules, de gaules, de perches et de recrutés de l'espèce spécifiée. La chaîne de caractères est destinée à être imprimée dans le fichier de sortie 'juvenile_libelleEspece'.

NB. Toute la gestion des conversions unités réelles / unités pixels et inversion de l'axe des y est réalisée par EnvironmentForest. Les méthodes de Placeau se contentent de réaliser des tests et des calculs en coordonnées "placeau", ou cartésiennes (en m).

Variables d'instance.

- *arbres* (équivalent à *listeArbres*) : liste des arbres vivants de Placeau. Mise à jour à la demande de EnvironmentForest à travers les méthodes : **addTree: aAgent voisins: booléen** et **mort: unArbre**.

- *arbresANaitre* : tableau contenant des couples (CODESS, nbRecrutés), stockant le nombre des arbres qui doivent être recrutés en fin de cycle. Elle est mise à jour par la méthode **choisiTonDestin: executerStadeJuvenile**, puis écrasée en fin de cycle après que tous les recrutés aient été transformés en nouveaux arbres par **effectTonDestin: cycleCourrant** (la variable *cycleCourrant* sert à attribuer une "date de naissance" au nouveau recruté).

- *arbresMorts* : liste des arbres morts de Placeau. Mise à jour à la demande de EnvironmentForest à travers les méthodes : **mort: unArbre** et **oubliLeMort: unMort**. Cette dernière méthode est, je pense, utilisée pour nettoyer la liste des morts à intervalle régulier spécifié au niveau de FenetreMain par l'utilisateur (à travers la variable *cyclesSauve*, qui va elle-même renseigner la variable *nbCyclesAvantOubli* de EnvironmentForest).

- *accointance* (équivalent à *placeauxProches*) : liste des 8 placeaux proches du sujet, ordonnés comme décrit plus haut.

- *evolution* : dictionnaire de dictionnaires de dictionnaires stockant l'évolution des différents stades de la régénération sur le placeau. Il se présente de la manière suivante :

```
#cycle1
  #surfaceTerrierePonderee
  #espèce1
    #graines
    #plantules
    #gaules
    #perches
    #recrutés
  #espece2
    #graines
    #plantules
    #gaules
    #perches
    #recrutés
...
#cycle2
  #surfaceTerrierePonderee
  #espèce1
    #graines
    #plantules
    #gaules
    #perches
    #recrutés
  #espece2
    #graines
    #plantules
    #gaules
    #perches
    #recrutés
#cycle3
...
```

Il est mis à jour par les méthodes **initEvolution** et **miseAJourEvolution** en ce qui concerne **#surfaceTerrierePonderee** et par la méthode **calculRecrutesEtMiseAJour: dicoCycle** de **StadeJuvenile** pour le reste.

- **placeauxLointains** : stocke l'ensemble des placeaux des 2ème et 3ème couronnes, dans l'ordre spécifié au niveau du schéma ci-dessus. Les 12 placeaux des coins sont stockés en fin de liste.

- **position** : variable héritée de la classe **Agent**. Coordonnées (en nombre de placeaux depuis l'origine 0@0 située en bas à gauche de l'écran) du placeau sujet : 0@0 à 24@24.

- **stadeJuvenile** : dictionnaire d'instances de **StadeJuvenile(Espece)**. Il est créé et renseigné par **SelvaOptions** à travers l'instruction : "**SelvaOptions dicoEspècesEtInstancesClassesJuveniles**". Celle-ci crée et initialise autant d'instances de la classe **StadeJuvenile(Espece)** (ex : **StadeJuvenileAngelique**) que d'espèces étudiées et les range dans **stadeJuvenile**, aux clés **#espèce1**, **#espèce2**, etc. Chaque instance possède les variables suivantes : **apportGraines**, **graines**, **plantules**, **gaules**, **perches** (variables de la superclasse **StadeJuvenile**).

- **surfaceTerriere** : ST totale en cm² sur le placeau.

- **surfaceTerrierePonderee** : calculée par la méthode **initializeCycle**. On somme les ST des 8 placeaux proches, on divise le résultat par 8, on l'ajoute à la ST totale du placeau sujet et on divise le tout par 2 ! Le calcul est fait en m² :

$$ST_{pondérée} = [ST_{sujet} + (ST_1 + \dots + ST_8)/8] / (2*10\ 000).$$

Variables de classe.

- **Representation** : il s'agit d'un rectangle d'origine 0@0 et d'extension 10@10, qui représente le placeau (mètres).

- **Size** : il s'agit de la dimension (en mètres) du placeau.

- **WorldDim** : Coin supérieur droit de la parcelle étudiée.

2.3.3 - Evolution

Il s'agit d'une sous-classe de **Model**. Une instance de cette classe est créée par **EnvironmentForest**, au moment de son initialisation : "**evolution:= Evolution new initialize**" dans **initialize: cellsSize**. Son rôle essentiel est de tenir à jour un dictionnaire : **tabMemoire**, qui contient tous les paramètres du peuplement à chaque cycle. C'est de **tabMemoire** que sont extraites les informations utilisées par **EvolutionView** et **EffectifsDiametreView** (méthodes "**tableauY**"). Lorsque **FenetreEvolution** et **FenetreDiametre** créent respectivement les instances des deux classes précédentes, elles leur passent une instance de **Evolution**.

tabMemoire est constitué de la manière suivante :

INFORMATIONS	Cycle -1	Cycle 0	Cycle 1	Cycle ...
#nbArbres				
#surfaceTerriere				
#nbMortsSurPied				
#nbMortsParChablis				
#nbMortsParChablisPrimaire				
#nbMortsParChablisSecondaire				
#nbNaissances				
#effectifParDiametre		0	10	15
		20	25	etc.
		95		
		10000		
#piedEtChablisParDiametre		etc.		
#chablisParDiametre		etc.		
#chablisPrimaireParDiametre		etc.		
#chablisSecondaireParDiametre		etc.		
#mortSurPiedParDiametre		etc.		

Toutes les informations sont ramenées à l'hectare. Les premières colonnes (cycle 0, cycle -1 et éventuellement cycle -2) sont remplies à partir des fichiers FIC.

A l'initialisation (au moment de la création de l'instance de Evolution), tabMemoire est créé en tant que dictionnaire. La taille d'un cycle est fixée à 3 ans et le tableau des effectifs par classes de diamètre comporte 20 colonnes d'amplitude 5 cm, sauf la première (0-10 cm) et la dernière (≥ 95 cm).

Lorsque la méthode "**postOpenWith: aBuilder**" de FenetreForet demande à EnvironmentForest "**lireForetInFichier**", "**lireForetInFichierFic**" est activé s'il n'y a pas lecture dans un fichier BOSS ; alors :

- "**evolution initializeAnnees: self noCampagne**" : initialise (en les mettant à zéro) les 7 premières variables du tableau ci-dessus, pour le cycle 0 et les cycles précédents ;
- "**evolution initializeParFichier: arrayTree**" : renseigne les 7 premières variables pour le cycle 0 et les cycles précédents, à partir des infos du fichier ; cette fonction est appelée 2 fois : une fois pendant la lecture du fichier des vivants, et une fois pendant la lecture du fichier des morts ;
- "**evolution reporterSuHectare: self**" : à la fin de la lecture du fichier, récupère les 7 variables renseignées et les reporte à l'hectare, pour le cycle 0 et les cycles précédents ;
- "**self initializeApresLesArbres**" >> "**evolutionInitializeCycleZero: self**" >> "**tabMemoire at: 0 put: (self initializeCycleParClasse: envt dico: dico**" : il y a rajout à tabMemoire des 6 dernières informations du tableau ci-dessus (infos par classes de diamètre). Les champs sont créés, initialisés à 0, puis remplis à partir des infos détenues par l'environnement (sous forme de listes) POUR LE CYCLE ZERO.

Par la suite, tabMemoire est renseigné à la fin de chaque cycle sur "ordre" de EnvironmentForest : méthode **act** >> "**evolution initializeCycle: self**". Un nouveau "dico" est ajouté à la clé cycleCourant..

Variables d'instance.

- *intervalsDiametre* : liste (de type tableau) contenant 20 éléments, à savoir les bornes des différentes classes de diamètre. Créée et initialisée lors de la création de l'instance de *Evolution*. Cette liste renseigne *EffectifsDiametreView* : méthode `tableauX >> "^self model tableauEffectifDiametreX" >> "^intervalsDiametre"`.
- *pasDeTemps* : initialisé à 3.
- *premierCycle* : défini dans la méthode `initializeAnnees` : `anneeDepart`. Partant de 0, la variable *cycle* est décrémentée pour initialiser *tabMemoire*.. En sortie de boucle : `"premierCycle:= cycle+1"`. Ce qui met la variable à 0.
- *tabMemoire* : dictionnaire créé à l'initialisation. Voir description plus haut.

2.3.4 - ForestFile

Sous-classe de *Object*. Une instance de cette classe est créée par *EnvironmentForest* dans `lireForetInFichierFic` : `"foretFichier:= ForestFile new initialize: 0 sizeWord: self worldRealDimension annee: (self noCampagne) fichier: (nomFichierIn, '.vivants')"`. NB : la même instruction revient par la suite pour le fichier d'extension '.morts'. Le chiffre 0 indique le nombre d'arbres que l'on veut charger au départ : il n'est différent de 0 que lorsqu'on ne charge qu'une partie de la forêt. "*noCampagne*" est lu dans l'en-tête du fichier FIC par *FenetreMain* qui le passe à *EnvironmentForest*. "*worldRealDimension*" vaut 100@100 dans le cas d'une parcelle de 1 ha. "*nomFichierIn*" a été sélectionné par l'utilisateur dans une liste, au niveau de *FenetreMain*.

L'initialisation renseigne les variables d'instance, ouvre le fichier et positionne un pointeur (*buffer*) sur le premier caractère. La fonction `readNextTree` génère une boucle de lecture qui permet de sauter les lignes correspondant à des arbres "non acceptables" (parce que situés en dehors du monde étudié) et de détecter une cause de fin de lecture. Lorsqu'une ligne valide est repérée, *dicoArbre* est rempli grâce à la fonction `readLine` qui décode le fichier caractère par caractère. Puis la main est rendue à l'appelant, ie *EnvironmentForest* (méthode `lireForetInFichierFic`) qui la redonne à la fin d'une boucle de travail concernant le nouvel arbre (méthode `nextTree`).

Remarque : lors de l'initialisation, un premier appel à `readNextTree` provoque le stockage des infos de la première ligne valide dans *dicoArbre*. Par la suite, `lireForetInFichierFic` "ordonne" : `"arrayTree:= foretFichier nextTree."` La fonction `nextTree` fait elle-même appel à `readNextTree` après recherche d'une fin de fichier, mais elle renvoie bien le vieux dicoArbre, mis de côté avant appel à `readNextTree`. Il y a donc lecture "en avance" des infos de l'arbre suivant.

Méthode importante : `readLine`. Le fichier déchiffré est un '.vivants' ou un '.morts' fabriqués par le programme *voisins.c* de D. Caruso. Les champs sont les suivants :

-	- PARC	- CARRE	— ARBRE	— CODE	— X	— Y	- NB MESURES
---	--------	---------	---------	--------	-----	-----	--------------

— DATE1	— CIRC1	— DATE2	— CIRC2	— ...	— DATEn	— CIRCn
---------	---------	---------	---------	-------	---------	---------

— NBD	-	— NBVloint.	-	— LVOIS1	— LVOIS2	-	— ...	— LVOISm
-------	---	-------------	---	----------	----------	---	-------	----------

Les infos sont stockées dans *dicoArbre*. Les dates de mesure ne sont pas lues. Un tableau de la taille de "NB MESURES" est créé pour stocker la série des circonférences (ramenées de mm à cm). Ce tableau est rattaché à la clé #circonference. NBD représente le nombre de voisins de diamètre supérieur à celui du sujet dans un rayon de 30m (calculé 3 ans avant la campagne de démarrage des simulations). NBVloint. est le nombre total de voisins (de toutes tailles) situés dans un rayon de 30m et non présents dans les placeaux proches du placeau auquel appartient l'arbre sujet. Les champs suivants stockent le numéro de ligne correspondant à chacun de ces voisins. Ils sont rassemblés dans un tableau de la taille NBVloint., rattaché à la clé #tabVoisins.

Le pointeur est ensuite repositionné à la ligne suivante. Une clé supplémentaire : #diametre est ajoutée au dictionnaire : elle indexe la liste des circonf. converties en diamètres. NB. Les diamètres sont stockés en cm.

Un test vérifie la validité des coordonnées de l'arbre. La mémoire est nettoyée toutes les 100 lignes.

ticoArbre est constitué de la manière suivante :

INFORMATIONS

#noParcelle		
#noCarre		
#noArbre		
#codeEspece		
#coordX		Abscisse convertie en mètres
#coordY		Ordonnée convertie en mètres
#nbMesures		
#noLigne		
#annee		Correspond à l'info. lue dans l'en-tête du fichier : numéro de la campagne de démarrage des simulations
#circonference	1	On accède à la circonférence voulue en utilisant le n° de la campagne. Ainsi " <i>ticoArbre at: (#circonference at: annee)</i> " retourne la circonférence à la campagne de démarrage.
	2	
	...	
	10	
#nbVoisinsTmoins1		Variable NBD calculée 3 ans avant la campagne de démarrage.
#tabVoisins	1	Numéros de ligne des voisins dans le fichier FIC d'origine.
	2	
	...	
	NBVloint	
#diametre	1	Les diamètres sont stockés en cm.
	2	
	...	
	10	

Variables d'instance.

- *annee* : passée par l'appelant (`EnvironmentForest : "self noCampagne"`).
- *buffer* : pointeur défini à l'initialisation.
- *compteur* : compte le nombre d'arbres réellement acceptés par l'environnement à chaque lecture de fichier.
- *ticoArbre* : stocke les infos contenues dans chaque ligne de fichier.
- *maxArbres* : passé en paramètre par l'appelant. Nombre maxi. d'arbres à charger dans l'environnement.
- *maxDistance* : taille du monde étudié (passé en paramètre par l'appelant). Permet d'éviter de charger les arbres dont les coordonnées sont invalides.
- *noLigne* : numéro de la ligne du fichier FIC où se trouve décrit l'arbre.

2.3.5 - ForestKernelScheduler

Sous-classe de `Object`. Une instance est créée par `EnvironmentForest` lors de son initialisation : `initializeScheduler >> "scheduler:= ForestKernelScheduler new environmentForest self"`. Cette instruction a notamment pour effet de créer un process actif, par la méthode `initializeProcess` :

```
process:= [
    [true] whileTrue: [
        self environmentForest act.
        self incAbsoluteCounter.
        Processor yield.
    ]
] newProcess.
```


L'instruction "*Processor yield*" permet de reprendre la main en cours de travail.

Les 3 méthodes importantes sont **run**, **suspend** et **terminate**. Les méthodes **run** et **suspend** sont indirectement "branchées" derrière les deux boutons superposés labellisés respectivement "*run*" et "*suspend*" de FenetreForet. En fait, elles sont activées par l'intermédiaire de deux méthodes de FenetreForet : **run** et **suspend**, qui rendent visibles ou invisibles certains composants de la fenêtre.

La méthode **terminate** est activée par un message de EnvironmentForest en fin de simulation : "*self changed: #terminate*" (voir méthode **act**), capté par ses dépendances et notamment FenetreForet.

Variables d'instance.

- *absoluteCounter* : value holder initialisé à 0 lors de la création de l'instance, et incrémenté de 1 à chaque fois que l'on relance le process.
- *environmentForest* : instance de EnvironmentForest, passée par elle-même au moment de l'initialisation du scheduler.
- *process* : défini à l'initialisation : voir son contenu ci-dessus.

2.4- Les arbres

2.4.1 - EspeceGrise

Sous-classe de Agent. Les instances de EspeceGrise sont créées par EnvironmentForest au moment du chargement du fichier '.vivants' (**lireForetInFichierFic** >> "*EspeceGrise new: code initializeWith: arrayTree voisins: booleen*") et par Placeau au moment du recrutement des arbres (**recrutement: cycleCourant codeEspece: code** >> "*EspeceGrise new: code initialize*"). "*new: code*" permet de demander à SelvaOptions de renvoyer le libellé correspondant au code et ainsi de renseigner la variable *libelleEspece*.

Avant création de ces instances, EnvironmentForest envoie à la classe EspeceGrise le message : **initializeClassVariables**. Cette méthode de classe initialise la variable de classe *AnneesParCycle* et crée une instance de la classe ModuleInstanceUnique : "*ModuleInstanceUnique initModules*". Cette dernière méthode a pour effet de créer et d'initialiser une instance de toutes les sous-classes, à savoir :

- *Deliberation* ;
- *DeliberationAngelique* ;
- *DeliberationGrise* ;
- *DeliberationPetitGris* ;
- *EffectionGrise* ;
- *EffectionAngelique* ;
- *EffectionPetitGris* ;
- *DetectionGrise*.

La méthode **initialize** (appelée directement par Placeau, ou par la méthode **initializeWith: arrayTree**) initialise toutes les variables d'un arbre recruté :

- *age*:= 0 ;
- *diametre*:= 10 ;
- *diametreT1*:= 9,7 ;
- *dateDeNaissance*:= 0 ;
- *etat*:= #vivant ;
- *listeVoisins*:= OrderedCollection new ;
- *evolution*:= Dictionary new (tableau [-1, -1, -1]) ;
- *libelleEspece*:= SelvaOptions codeEspeceToLibelleEspece: codeEspece.

Ces variables sont ensuite reprises lorsque l'initialisation doit être faite à partir d'une ligne du fichier '.vivants'. **NB.** L'appel à la méthode d'accès **diametre:** <> entraîne également le calcul de la variable *surfaceTerriere*. Le diamètre récupéré dans le fichier est "*(arrayTree at #diametre) at: (arrayTree at: #annee)*", #annee étant le numéro de la campagne de départ (*noCampagne* de EnvironmentForest). *diametreT1* concerne le cycle précédent. La date de naissance (première apparition d'un diamètre > 0 et < 999) est déterminée en nombre de cycles (négatif si avant la campagne de démarrage), sachant qu'un arbre naît toujours en fin de cycle : -> variable

dateDeNaissance. Si elle vaut 0, *nbVoisinsT1* est mis à -1. C'est la valeur de cette dernière variable qui permet de repérer un recruté et de le brancher sur des modules spécifiques. En cours de simulation, lors d'un recrutement, *dateDeNaissance* vaudra *cycleCourant* (affecté par Placeau).

La méthode **initializeModules**, appelée en fin de méthode **initializeWith: arrayTree** ou directement par Placeau lors d'un recrutement teste la valeur de *nbVoisinsT1* et branche l'instance soit sur **initializeJeune**, soit sur **initializeVieux**, qui permettent la connexion de l'arbre à ses 3 modules "de vie", *moduleDetection*, *moduleDeliberation* et *moduleEffection*, renseignés de la manière suivante :

- **initializeJeune**

moduleDetection:= DetectionGrise instance.
moduleDeliberation:= DeliberationPetitGris instance.
moduleEffection:= EffectionPetitGris instance.

- **initializeVieux**

/ classeDeliberation /
classeDeliberation:= SelvaOptions libelleEspeceToClasse: libelleEspece.
moduleDetection:= classeDeliberation detectionAssociee.
moduleDeliberation:= classeDeliberation instance.
moduleEffection:= classeDeliberation effectionAssociee.

Les modules de "délibération" et d'"effection" des jeunes ne leur sont affectés que pour un cycle. Dès le cycle suivant, EffectionPetitGris ordonne à l'arbre qui lui a été passé : **initializeVieux**. La méthode **libelleEspeceToClasse: libelleEspece** de SelvaOptions va déterminer à quel sous-module de Deliberation il faut connecter l'arbre, ce en fonction de son espèce d'appartenance.

La méthode **initializeAvecEnvironment: arbres** est appelée par EnvironmentForest dans **lireForetInFichierFic**, après la lecture du fichier '.vivants'. Elle permet de constituer, pour chaque arbre, la liste *listeVoisins* à partir des infos contenues dans #tabVoisins (voir ForestFile). Les voisins concernés sont les arbres de toute taille situés dans un rayon de 30 m et en dehors de l'accointance.

La méthode **initializeAvecEnvironment** est ensuite activée (elle l'est également directement par Placeau, en fin de méthode **recrutement: <> codeEspece: <>**). Elle a pour effet de renseigner la variable *noCarre* en fournissant à EnvironmentForest les coordonnées de l'arbre (c'est important pour les nouveaux recrutés !), puis de créer une nouvelle instance de TableauEvolutionArbre en lui fournissant les infos *diametre*, *nbVoisinsT1* et "-1". Cette instance est ajoutée au dico. *evolution* à la colonne "*self environment cycleCourant*" (méthode **evolutionAdd: <>**). A priori, en fin de lecture du fichier '.vivants', la simulation n'a pas encore démarré donc *cycleCourant* vaut 0, donc *evolution* est renseigné à partir de 0.

2.4.1.1 - Le point sur : le destin des arbres.

Dans le courant de la simulation, se produit un va-et-vient permanent entre EspeceGrise, ses modules de vie, EnvironmentForest et Placeau. Dans un premier temps, EnvironmentForest demande aux arbres de délibérer ("*self deliberationArbres*"). La méthode **choisiTonDestin** est activée et l'arbre s'adresse à son module de délibération ("*doAct: self*", méthode de la superclasse Deliberation) : le but est de remplir sa variable-tableau *destin*.. Différents résultats peuvent être obtenus :

(1)	#croissance	ΔD	NBD	
(2)	#croissance	ΔD	NBD	#disseminationGraines
(3)	#croissance	ΔD	NBD	#pasDeDisseminationGraines
(4)	#mortSurPied			
(5)	#mortParChablisPrimaire	direction de chute		
(6)	#mortParChablisSecondaire	position du chablis primaire		
(7)	#mortParChablisMultiple	direction de chute (= vers chablis primaire ou dans la même direction)		

(8)	#mortParEclaircie
-----	-------------------

Avant toute chose, la variable *evolutionFuture* est créée (instance de TableauEvolutionArbre) et initialisée. Puis le module de délibération teste dans l'ordre :

- la possibilité d'occurrence d'un chablis primaire (à partir du diamètre courant et de la surface terrière des voisins de toutes tailles dans un rayon de 30 m). Si le chablis doit intervenir, une direction de chute (entre 1 et 8) est tirée au hasard et le tableau (5) est retourné ;
- la possibilité d'occurrence d'une mort sur pied (à partir du diamètre courant et du diamètre au cycle précédent : *diametreT1*). Si la mort doit intervenir, le tableau (4) est retourné ;
- si ces deux étapes sont franchies sans encombre, la fonction **calculCroissance: moduleAgent** des sous-classes de Deliberation est appliquée à l'arbre. **ATTENTION** : le fonctionnement est différent selon que l'arbre est rattaché à DeliberationGrise ou à Deliberation(Espece). Dans le premier cas, la variable *evolutionFuture* est renseignée avec [diamètre courant + ΔD , NBD au cycle courant, -] et le couple [ΔD , NBD au cycle courant] est renvoyé à la fonction appelante. Celle-ci retourne alors le tableau (1). Dans le deuxième cas, par exemple pour DeliberationAngelique, le module récupère d'abord auprès de son supérieur hiérarchique (Deliberation) les infos précédentes (les actions sont les mêmes) puis, si l'arbre possède la clé #croissance et si la variable *stadeJuvenile* de EnvironmentForest a été mise à *true* par l'utilisateur, alors il interroge l'arbre sur son intention de fleurir. L'arbre doit fleurir : la fonction retourne un tableau de type (2) et renseigne la liste *listeArbresDisseminateurs* de EnvironmentForest. L'arbre ne doit pas fleurir : la fonction retourne un tableau de type (3).

NB. Pour un cycle donné du tableau *evolution* attaché à chaque arbre, le triplet [diamètre, NBD, STV] contient le diamètre atteint par l'arbre en fin de cycle et la valeur que l'indice de comp. NBD avait au début du cycle (*ie en fin de cycle précédent, ie celle qui a permis de calculer le diamètre du triplet* !).

La main étant rendue à l'arbre, celui-ci vérifie si *destin* est de type (5). Si c'est le cas, il appelle son module d'effection et lui demande successivement : "*propageChablisPrimaire: self directionDeChute: (destin at:2)*" et "*propageChablisMultiple: self directionDeChute: (destin at:2)*".

- Dans le premier cas, la fonction repère les arbres qui sont susceptibles d'être touchés par le chablis primaire et elle leur demande d'examiner cette éventualité en leur fournissant une probabilité de mourir ainsi qu'un nombre aléatoire. Si les arbres "décident" de mourir (il faut d'abord que leur *destin* soit vide, ou contienne #croissance), ils affectent eux-mêmes à *destin* un tableau de type (6).
- Dans le deuxième cas, la fonction n'a d'effet que si le chablis primaire est suffisamment gros : diamètre courant ≥ 40 cm. Si c'est le cas, EffectionGrise demande aux arbres du plateau porteur d'envisager la mort par chablis multiple dans la même direction que celle prise par le chablis primaire. Pour cela, les arbres visés s'adressent à leur module de délibération ("*mortParChablisMultiple: self*") qui prend une décision en fonction de la surface terrière totale des voisins (dans un rayon de 30 m). Si la mort doit se produire, l'arbre affecte lui-même à sa variable *destin* un tableau de type (7), la direction de chute étant la même que celle du chablis initial.

Ensuite, EffectionGrise s'adresse aux 8 placeaux proches et propose à chacun de leurs arbres de tomber dans la direction du plateau initial (= porteur du chablis primaire). La méthode utilisée est la même que précédemment ("*unArbre propositionDeMortParChablisMultiple: direction*").

- Dans tous les cas, lorsqu'un arbre doit tomber à la suite d'un chablis complexe, il appelle de nouveau les deux méthodes "*propageChablisPrimaire: self directionDeChute: direction*" et "*propageChablisMultiple: self directionDeChute: direction*" de son module d'effection. Une boucle est ainsi enclenchée, sans critère d'arrêt. L'arrêt se produit du fait de probabilités de plus en plus faibles d'occurrence.

Cette propagation possible des chablis primaires fait que lorsque l'on demande pour la première fois à un arbre **choisiTonDestin**, sa variable *destin* peut déjà être renseignée, auquel cas il ne fait rien.

Lorsque tous les arbres ont été passés en revue, EnvironmentForest demande aux disséminateurs de disséminer, puis aux placeaux de délibérer sur le recrutement. Enfin, il active **effectionArbres**. Les arbres sont pris les uns après les autres et on leur demande : **effectTonDestin**. Chacun appelle alors son module d'effection ("*^self moduleEffection doAct: self with: destin*"), puis met sa variable *destin* à 0.

La méthode **doAct: moduleAgent with: anAct** de EffectonGrise (voir paragraphe concerné) examine le premier champ de *destin* et active toutes les fonctions correspondantes. Dans tous les cas de mort, la variable *etat* de l'arbre est renseignée. Initialisée à #vivant, elle devient selon le cas :

- #mortSurPied,
- #mortParChablisPrimaire,
- #mortParChablisSecondaire,
- #mortParChablisMultiple,
- #mortParEclaircie.

Ensuite, le placeau propriétaire est averti par 3 fonctions possibles : **mortParChablis: moduleAgent**, **mortSurPied: moduleAgent**, **mortParEclaircie: moduleAgent** qui ont toutes le même effet à savoir (a) décrémenter la ST du placeau ; (b) alimenter la liste *arbresMorts* du placeau ; (c) avertir EnvironmentForest (**mortArbre: unArbre**. EF décrémente alors *nbArbres* et *surfaceTerriere*, incrémente *nbMortsInCycle*, alimente sa liste *arbresMorts*) ; (d) retirer l'arbre mort de la liste *listeArbres* du placeau.

Enfin, l'arbre mort est retiré de la liste *listeVoisins* de tous ses voisins lointains, sa propre liste de voisins est mise à 0 et sa variable *dateDeMort* prend la valeur "*environment cycleCourant*".

Si le destin de l'arbre est de pousser, le diamètre courant est passé à *diametreTI*, le nouveau diamètre est calculé ainsi que la nouvelle surface terrière et un message est envoyé au placeau propriétaire : **incSurfaceTerriere: <>**. Cette méthode recalcule la ST du placeau, demande à EnvironmentForest de recalculer la ST totale, puis avertit les dépendances du placeau : "*self changed: #croissance with: aTree*". PlaceauView capte cet ordre à travers **update: <> with: <>** mais il n'a pas d'effet : le seul changement susceptible de provoquer un réaffichage par cette méthode est le cas : #mort. Ensuite, la variable *nbVoisinsTI* de l'arbre est renseignée par *destin* (champ n°3) et la méthode **evolutionAdd** est activée. Elle a pour effet d'ajouter au tableau *evolution* de l'arbre, le tableau *evolutionFuture*. à l'"entrée" "*environment cycleCourant*". *evolutionFuture* semble avoir un champ toujours vide (le 2ème ou le 3ème en fonction du module de délibération qui a été utilisé).

2.4.1.2 - Le point sur : le cas des éclaircies.

Avant le démarrage de la simulation, l'utilisateur peut agir sur la forêt en demandant une éclaircie (menu "Eclaircie" de FenetreForet). On va alors générer un cycle fictif au cours duquel les arbres ne pousseront pas, ne se reproduiront pas, mais pourront éventuellement mourir.

- Activation de la méthode **eclaircieSystematique** de FenetreForet. Un message est envoyé à EnvironmentForest : "*self owner éclaircieSystematique*" qui prend un à un ses arbres de *listeArbres* et leur demande **compteToiSiEclaircie**. L'arbre compare alors son diamètre à la valeur "*diamCritique:= SelvaOptions diametreCoupe: self libelleEspece*" et renvoie un booléen. S'il est vrai, EF incrémente un compteur (dont la valeur apparaît dans une fenêtre de dialogue). Si l'éclaircie est confirmée par l'utilisateur, la méthode de EF **eclaircie: #systematique** est activée. Il s'agit d'un équivalent de la méthode **act** : de nombreux ordres sont communs à ces deux méthodes. *cycleCourant* est incrémenté de 1, sans que cyclesRestantsAEffectuer soit décrémenté (→ cycle fictif). **inititalizeCycle** met à 0 les variables *nbMortsInCycle* et *nbNaissancesInCycle* et à -1 *surfaceTerrierePondereeMaxi..* On passe ensuite à chaque arbre le message **choisiTonEclaircie: #systematique** qui a pour effet de demander au module de délibération : "*doEclaircie: self avecCritere: #systematique*". Cette dernière fonction (de la superclasse Deliberation) **refait le test de comparaison des diamètres** (ATTENTION : il faudrait là ré-utiliser la fonction **compteToiSiEclaircie** de l'arbre, ce serait plus propre !!!). Si l'arbre doit mourir, un tableau de type (8) est renvoyé pour renseigner le *destin* de l'arbre. Dans le cas contraire (l'arbre reste vivant), le tableau renvoyé est de type (1) avec $\Delta D = 0$.

EF demande ensuite à l'arbre : **effectTonDestin** et la série des actions réalisées par EffectonGrise est la même que pour tous les arbres morts (voir description plus haut). La variable *etat* de l'arbre devient #mortParEclaircie.

A la fin de ces instructions, la liste *arbresMorts* de EF se trouve remplie. EF les prend un à un et les élimine de sa liste d'arbres vivants (*listeArbres*).

Une instruction **inititalizeCycle: self** envoyée à Evolution permet de mettre à jour son dico. *tabMemoire*, puis les placeaux reçoivent l'instruction **evolutionEclaircie** qui leur permet de mettre à

jour leurs tableaux *evolution*: (un nouveau dico. est ajouté à l'entrée "*self owner cycleCourant*" avec mise à -1 de *#surfaceTerrierePonderee*) et *stadeJuvenile*. Enfin, l'instance d'EnvironmentForest indique à ses dépendances qu'un nouveau cycle va démarrer : "*self changed: #nouveauCycle*".

- Activation de la méthode **eclaircieSelective** de FenetreForet. Une nouvelle instance de FenetreEclaircie est créée. Elle s'initialise sur le premier plateau de la liste *cells* de EF. Tout changement de "selectionIndexHolder" sur *listeNomsArbres* (instance de SelectionInList, variable de FenetrePlaceau qui double la variable *listeArbres*) provoque l'activation de la méthode **changeListeNomsArbres** de FenetreEclaircie. L'arbre sélectionné (soit directement dans la liste, soit par cliquage de la souris sur la carte) est testé : s'il est vivant, un compteur est incrémenté et le message **tuEsSelectionnePourEclaircieSelective** lui est envoyé : il fait passer *etat* de *#vivant* à *#mortParEclaircie*. Si l'arbre avait déjà été sélectionné, le compteur est décrémenté et la même méthode de EspecesGrise fait passer *etat* de *#mortParEclaircie* à *#vivant*. A chaque fois se produit un réaffichage de l'arbre dans la vue.

Lorsqu'on est satisfait des sélections, on active, dans FenetreEclaircie, la méthode **lancerEclaircie**, qui envoie à EF le message "*eclaircieSelective: (nbArbresSelectionnees value)*". Si l'utilisateur confirme son intention, la méthode **eclaircie: #selective** est activée à son tour. A partir de là, le déroulement des actions est le même que pour l'éclaircie systématique, si ce n'est que Deliberation ne teste plus la valeur du diamètre de l'arbre.

NB. On peut faire ce genre d'intervention à la fin de n'importe quel cycle.

2.4.1.3 - Le point sur : le cas des recrutés.

Une fois que la méthode **effectationArbres** a été activée par EnvironmentForest et que la main lui a été rendue, on passe à **effectationPlaceaux**. Les placeaux procèdent alors au recrutement, en fonction des infos fournies par la liste *arbresANaitre*. De nouvelles instances de EspecesGrise sont créées et initialisées, auxquelles on demande notamment : "*arbre initializeModules*", "*arbre naissance*" et "*arbreInitializeAvecEnvironment*". Le nouvel arbre est connecté à ses modules de vie spécifiques, il demande à DetectionGrise de lui fabriquer sa liste de voisins lointains et de l'ajouter dans *listeVoisins* de tous ses voisins lointains (méthode **naissance: agentArbre**), on recherche son carré d'appartenance et on renseigne son tableau *evolution* avec une instance de TableauEvolutionArbre qui contient : [diametre (=10 cm), nbVoisinsT1 (= -1), -1]. *nbVoisinsT1* vaut -1 car non renseigné.

Donc, ces arbres n'interviendront dans la vie de la parcelle qu'au cycle suivant, au travers de DeliberationPetitGris et EffectationPetitGris (voir plus bas les chapitres correspondants).

2.4.1.4 - Autres méthodes et variables d'instance.

- **couleur** : appelée par Placeau (méthode **displayTree: aTree on: aView**). Cette méthode teste la variable *etat* de l'arbre et va chercher dans le dico *couleurs* de SelvaOptions la couleur correspondante pour l'affichage.
- **randomAutocorrele** : méthode appelée par Deliberation (**random: moduleAgent**). Elle va chercher dans SelvaOptions le booléen entré par l'utilisateur pour l'espèce désignée par "*self libelleEspeces*" et qui permet d'introduire une corrélation entre deux valeurs successives issues du tirage au hasard.
- **codeNbMortToSymbol: code** : appelée par Evolution (**initializeParFichier: arrayTree**). Elle permet de traduire en symboles les codes numériques de mortalité contenus dans les fichiers FIC. Ces symboles sont ensuite utilisés comme clés dans *tabMemoire*.
- **toutesDeliberationsEspeces** : méthode appelée par FenetreEspeces. Elle retourne "**^Deliberation allSubclasses**", sauf DeliberationPetitGris et DeliberationEclaircie.

Variables d'instance.

- **age** : stocke l'âge de l'arbre. Initialisée à 0 dans **initialize**. Cette variable est incrémentée par la méthode **evolutionAt: cycle add: x** appelée soit par EffectationGrise ("**doAct: moduleAgent with: anAct**") si l'arbre a participé activement au cycle, soit par **initializeAvecEnvironment** si l'arbre vient d'être recruté en cours de cycle (ou lu dans le fichier '.vivants').

- *codeEspece* : renseigné par le champ #codeEspece de *dicoArbre* (tableau de ForestFile). Pour ce qui est du recrutement (création de nouvelles instances d'EspeceGrise), le code est affecté au moment de la constitution de sa liste "*arbresANaitre*" par Placeau (**choisiTonDestin: executerStadeJuvenile**).
- *dateDeNaissance* : initialisée à 0 et re-calculée à partir des infos contenues dans le fichier '.vivants', lors du premier chargement des arbres. Origine du calcul : la campagne de démarrage.
- *dateDeMort* : variable renseignée par le module d'effection : méthode **mort: moduleAgent** de EffectionGrise. "**moduleAgent dateDeMort:= moduleAgent environment cycleCourant**".
- *destin* : tableau rempli par le module de délibération pour renseigner le module d'effection. Elle permet le marquage d'un temps entre l'activation des 2 modules pour que tous les arbres agissent sur les mêmes bases de départ). 8 possibilités pour *destin* : voir tableau ci-dessus.
- *diametre* : stocke le diamètre de l'arbre calculé pendant le cycle en cours (ie valeur figurant dans le tableau *evolutionFuture* !). Cette valeur deviendra *diametreT1* au cours du cycle suivant (au moment du passage dans le module Deliberation).
- *diametreT1* : cette variable est renseignée en début de cycle avec la valeur de *diametre*, au moment du passage dans le module Deliberation.
- *etat* : "étiquette" initialisée à #vivant. Elle peut prendre différentes valeurs :

```
#mortSurPied,
#mortParChablisPrimaire,
#mortParChablisSecondaire,
#mortParChablisMultiple,
#mortParEclaircie.
```

Ces valeurs sont attribuées par le module d'effection.

- *evolution* : dictionnaire de tableaux qui sont des instances de TableauEvolutionArbre. Il est renseigné ("*evolution at: (self environment cycleCourant) put: evolutionFuture*") par la méthode **evolutionAdd** en fin de cycle, à la demande du module d'effection.
- *evolutionFuture* : instance de TableauEvolutionArbre, créée par **doAct: moduleAgent** de Deliberation dans le but de stocker les infos importantes concernant la croissance de l'arbre et d'alimenter le tableau *evolution*. Elle est renseignée au moment de l'appel à la fonction de calcul de la croissance, particulière à chaque sous-classe (DeliberationGrise ou Deliberation(Espece)). **ATTENTION** : il semblerait que tous les champs de *evolutionFuture* ne soient pas systématiquement remplis. Lorsque le module actif est DeliberationGrise, seules les variables *diametre* et *nbVoisins* (ie les 2 premiers champs) sont renseignés. Lorsque c'est DeliberationAngelique qui intervient, seules les variables *diametre* et *surfaceTerriere* (ie les champs 1 et 3) sont renseignés. **Il s'agit des variables intervenant dans le modèle de croissance. NB.** Dans tous les cas, *nbVoisins* et *surfaceTerriere* ont pour valeur celle qui a permis de calculer *diametre* ! C'est-à-dire, si l'on veut, que *diametre* caractérise l'arbre en fin de cycle, alors que les deux autres variables datent du cycle précédent.
- *libelleEspece* : au moment de l'instanciation d'EspeceGrise, le code espèce est passé en paramètre à la méthode de classe **new: codeEspece**. La méthode **initialize** donne ensuite comme instruction : "**libelleEspece:= SelvaOptions codeEspeceToLibelleEspece: codeEspece**".
- *ligneFichier* : n° de la ligne du fichier '.vivants' à laquelle ont été lues les infos caractérisant l'arbre. Valeur récupérée à partir du champ #noLigne de *dicoArbre* (tableau de ForestFile).
- *listeVoisins* : stocke la liste des voisins lointains (= leur identité) fabriquée par **initializeAvecEnvironment: listeVivantsEnvironment** à partir de *listeVoisinsLignes*.
- *listeVoisinsLignes* : renseigné au démarrage à partir du champ #tabVoisins de *dicoArbre* (tableau de ForestFile). Remis à "nil" une fois que les infos ont été exploitées pour constituer *listeVoisins*.
- *InRandomOld* : cette variable conserve la valeur du précédent nombre tiré au hasard, pour que l'on puisse introduire une corrélation entre valeurs aléatoires successives (cas où le booléen *randomAutocorrele* stocké dans SelvaOptions pour l'espèce considérée est à *true*).
- *moduleDeliberation* : renseigné par **initializeModules**. Stocke l'instance de Deliberation associée à l'espèce concernée (DeliberationPetitGris si l'arbre a été recruté au cycle précédent).

- *moduleDetection* : renseigné par **initializeModules**. Stocke l'instance de Detection associée à l'espèce concernée.
- *moduleEffection* : renseigné par **initializeModules**. Stocke l'instance de Effection associée à l'espèce concernée (EffectionPetitGris si l'arbre a été recruté au cycle précédent).
- *nbVoisinsT1* : nombre total de voisins "lointains" (ie non situés sur les placeaux de l'accointance) de toutes tailles. Initialisée à partir du champ #nbVoisinsTmoins1 de *dicoArbre* (tableau fabriqué par ForestFile). Il s'agit d'une info. collectée 3 ans avant le moment où l'on s'intéresse au diamètre. Dans DeliberationGrise ou DeliberationLog, cette variable sert à calculer l'IC ($\text{deltaVoisins} := \text{voisinsTmoins1} - \text{voisinsTmoins2}$) en renseignant le calcul de voisinsTmoins2 (méthode **nombreDeVoisins**: **moduleAgent** de DetectionGrise).
- *noCarre* : information récupérée du champ #noCarre de *dicoArbre* (tableau de ForestFile) pour les arbres préexistants, ou bien calculée par **initializeAvecEnvironment** à partir des coordonnées de l'arbre, pour les recrutés.
- *piedDuChablis* : position du chablis primaire qui a entraîné l'arbre dans un chablis secondaire. L'info. est stockée pour l'instruction de tracé (visualisation graphique) donnée par Placeau.
- *surfaceTerriere* : surface terrière de l'arbre. Mise à jour en même temps que le diamètre (par la méthode d'accès **diametre**: <>).

2.4.2 - TableauEvolutionArbre

Sous-classe de Array. Des instances de cette classe sont créées, à chaque cycle, par chaque instance de EspecceGrise pour alimenter la variable-tableau *evolution*.

Chaque instance est un tableau de 3 cases :

Diamètre	Indice NBD	ST des voisins
(-1)	(-1)	(-1)

Les méthodes d'accès sur chaque variable sont : **diametre**, **nbVoisins** et **surfaceTerriere**. Toutes les cases sont mises à -1 à l'initialisation.

2.4.3 - ModuleInstanceUnique

Sous-classe de "Object". Une instance de cette classe est créée par la méthode de classe **initializeClassVariables** de EspecceGrise. Le message **initModules** qui est alors envoyé provoque l'instanciation de toutes les sous-classes de ModuleInstanceUnique, chacune étant responsable d'une partie de la vie des arbres. Chaque nouvelle instance de EspecceGrise se rattachera à 3 d'entre elles (Detection(Especce), Deliberation(Especce) et Effection(Especce)).

Variables de classe.

- *monInstance* : renseignée (par "self") au moment de l'instanciation.

2.4.4 - DetectionGrise

Sous-classe de ModuleInstanceUnique. Une instance est créée au moment de l'appel à la superclasse. Elle renseigne la variable *moduleDetection* de EspecceGrise.

Ce module permet à l'arbre d'analyser son environnement : c'est lui qui fournit les éléments de calcul des différents indices de compétition au travers des méthodes suivantes.

- **listeCompleteDeVoisins**: **agentArbre**. Tous les voisins (proches et lointains) de l'arbre sont identifiés et une liste des arbres de taille supérieure ou égale à la sienne est constituée.
- **nombreDeVoisins**: **agentArbre** : retourne l'indice NBD.
- **surfaceTerriereVoisins**: **agentArbre** : retourne la surface terrière totale dans un rayon de 30 m (indice STT).

Ces méthodes sont mises en oeuvre par Deliberation, DeliberationGrise et Deliberation(Espece).

Par ailleurs, DetectionGrise constitue la liste des voisins lointains (*listeVoisins*) des nouveaux recrutés et inclut ces recrutés dans la liste *listeVoisins* de leurs voisins lointains : méthode **naissance: agentArbre**.

2.4.5 - Deliberation

Sous-classe de ModuleInstanceUnique et superclasse de DeliberationGrise, Deliberation(Espece) et DeliberationPetitGris. Elle s'occupe, pour tous les arbres, de "déliberer" sur la mortalité : sur pied, par chablis primaire, par chablis secondaire et par chablis multiple, en mettant en oeuvre des modèles communs à tout le peuplement. Elle s'occupe aussi de l'éclaircie systématique : test de la valeur du diamètre par rapport au diamètre critique de coupe affecté à une espèce (consultation de SelvaOptions).

Deliberation s'occupe de renseigner le tableau *destin* de chaque arbre connecté (méthode **doAct: unAgent**) : il peut renvoyer des tableaux de type (1), (4), (5) et (8) (voir § 2.4.1). Les tableaux (2) et (3) sont renvoyés par les sous-classes de Deliberation (Deliberation(Espece)) et les tableaux (6) et (7) sont retournés par EspeceGrise après consultation de Deliberation.

La mise en oeuvre du modèle de croissance est de la responsabilité des sous-classes : DeliberationGrise, Deliberation(Espece) et DeliberationPetitGris.

Par ailleurs, Deliberation gère l'autocorrélation entre résidus du modèle de croissance (effective lorsque "*moduleAgent randomAutocorrele*" est à *true*). A noter qu'elle crée une instance de Gauss1 pour le calcul de ces résidus (méthode **initialize**). Les sous-classes, au moment du calcul de la croissance, font appel à leur superclasse pour obtenir ces valeurs.

La méthode **initializeModules** >> **initializeVieux** détermine avant tout quel module de délibération doit être affecté à l'arbre. Les modules de détection et d'effection rattachés le sont ensuite par chaque module de délibération spécifique. Par défaut, il s'agit des instances de DetectionGrise et d'EffectionGrise. Ils sont communs à tous les arbres et se contentent, le premier d'analyser l'environnement et le deuxième d'appliquer les décisions de Deliberation contenues dans la variable *destin*.

Deliberation et ses sous-classes sont mis en oeuvre par **act** >> "*self deliberationArbres*" de EnvironmentForest.

Variables d'instance.

- *chaA, chaB, chaC, chaD* : paramètres du modèle de mortalité par chablis primaire (voir valeurs dans **initialize**).
- *pa, pb* : paramètres du modèle de mortalité sur pied (voir valeurs dans **initialize**).
- *randomGauss* : instance de la classe Gauss1. Intervient dans le calcul des résidus autocorrélés.

2.4.6 - DeliberationGrise

Sous-classe de Deliberation, instanciée au moment de l'appel à la superclasse ModuleInstanceUnique (voir § 2.4.3). Elle renseigne la variable *moduleDetliberation* de EspeceGrise. Elle possède bien évidemment toutes les propriétés de la classe précédente et elle gère la croissance des arbres de l'espèce indifférenciée.

Les IC sont calculés par appel à DetectionGrise. La fonction principale : **calculCroissance: moduleAgent** renvoie à Deliberation un tableau (ΔD , NBD(Tmoins1)) qui sera transmis à l'arbre pour son tableau *evolutionFuture*.

Variables d'instance.

- *ca, cb, cc, cd, cf, cg, ch, ci* : paramètres du modèle de croissance.

2.4.7 - DeliberationAngelique

En l'état actuel, il s'agit d'une sous-classe de Deliberation, instanciée au moment de l'appel à la superclasse ModuleInstanceUnique (voir § 2.4.3), mais en fait elle devrait hériter de DeliberationGrise (modif. à effectuer).

Elle renseigne la variable *moduleDeliberation* de EspeceGrise. Elle possède bien évidemment toutes les propriétés de la classe précédente et elle gère la croissance des Angéliques ainsi que leur aptitude à la reproduction.

Les IC utilisés dans le modèle de croissance sont différents de ceux de l'espèce commune => *evolutionFuture* est renseignée différemment. Lorsque le modèle spécifique est actif, il renvoie (*nouveauDiametre*, *STVoisinsTmoins1*).

La méthode **fleuri: moduleAgent** teste le diamètre de l'arbre et décide de son caractère de disséminateur au vu de cette variable. Elle renvoie un booléen.

Enfin, la méthode **doAct: moduleAgent** de Deliberation est redéfinie à ce niveau. Si l'arbre est déclaré en croissance (consultation de la superclasse), elle récupère et réécrit son tableau *destin*, pour renvoyer un tableau de type (2) ou (3) (voir § 2.4.1) signalant le caractère éventuellement disséminateur de l'arbre.

A noter que le module d'effection rattaché est spécifique (instance de EffectionAngelique), et qu'un module "juvenileAssocie" vient s'ajouter aux 3 autres.

Variables d'instance.

- *ca, cb, cc, cd, cg* : paramètres du modèle de croissance spécifique de l'Angélique.

2.4.8 - DeliberationPetitGris

En l'état actuel il s'agit d'une sous-classe de Deliberation, instanciée au moment de l'appel à la superclasse ModuleInstanceUnique (voir § 2.4.3). Elle devrait en fait hériter de DeliberationGrise (modif. à effectuer). Elle renseigne la variable *moduleDeliberation* de EspeceGrise, uniquement lorsque la méthode **initializeJeune** est appelée (cas d'un arbre recruté en fin de cycle précédent) et uniquement pour un cycle.

Deux méthodes :

- **calculCroissance: moduleAgent**. Elle affecte d'office au recruté une croissance de 0,3 cm par an.

- **doAct: moduleAgent**. Elle initialise une instance de TableauEvolutionArbre (-> *evolutionFuture*), et renvoie un tableau de type (1) (voir § 2.4.1) pour renseigner la variable *destin*. Donc, un jeune recruté est toujours déclaré en croissance lors du premier cycle. Il peut cependant être victime d'un chablis primaire (devenir chablis secondaire ou chablis multiple).

2.4.9 - EffectionGrise

Sous-classe de ModuleInstanceUnique, instanciée au moment de l'appel à la superclasse (voir § 2.4.3). Elle renseigne la variable *moduleEffection* de EspeceGrise, pour l'espèce commune. Son rôle consiste à mettre en application les décisions des modules de délibération, contenues dans la variable *destin* de chaque arbre. Cette mise en application est réalisée par la méthode **doAct: moduleAgent with: anAct**. EffectionGrise est la superclasse de EffectionAngelique et de EffectionPetitGris.

Lorsque le destin de l'arbre est de mourir, les étiquettes *etat* sont modifiées, les voisins de l'arbre sont avertis, ainsi que Placeau et EnvironmentForest qui mettent alors à jour leurs propres variables.

Cette classe est par ailleurs chargée d'examiner la propagation des chablis. Lorsqu'à la suite d'une délibération, un arbre se rend compte que le premier champ de sa variable *destin* vaut #chablisPrimaire, il adresse à son module d'effection deux messages :

- **propageChablisPrimaire: moduleAgent directionDeChute: direction** : a pour effet de créer des chablis secondaires, selon les règles suivantes. (a) Seuls les gros arbres créent des chablis secondaires. Le nombre d'arbres qu'ils peuvent casser varie selon leur taille : les règles sont contenues dans le tableau *tabPropagationChablisSelonDiametre* (voir ci-dessous). (b) Une fois déterminé le nombre de chablis secondaires à produire (*nbChablis*), on s'adresse aux deux placeaux des 1ère et 2ème couronnes situés dans la direction indiquée et on récupère tous les arbres de taille inférieure ou égale à celle du chablis primaire qui seront les candidats à la chute. On renvoie à chacun de ces arbres une probabilité de chute calculée comme suit : "**probabiliteDeChute := nbChablis / nbArbresCandidats**". (c) Si leur destin n'est pas déjà de mourir, ces arbres transmettent la variable *probabiliteDeChute* au module de délibération. (d) Ce dernier renvoie à l'arbre un booléen après comparaison de la proba. avec un nombre tiré au hasard.

- **propageChablisMultiple: moduleAgent directionDeChute: direction** : a pour effet de créer des chablis "complexes", en partant de l'hypothèse d'un déséquilibre du peuplement environnant. Donc, 1) ce déséquilibre n'est sensible que pour des gros chablis primaires : en pratique de plus de 40 cm de diamètre. 2) Il ne concerne que les arbres assez proches, ie ceux qui sont situés sur le plateau lui-même et sur ses "accointances". On procède de la manière suivante : (a) on propose aux arbres du plateau d'origine de tomber dans la même direction que le chablis primaire : pour cela, il faut repasser par l'arbre et par son module de délibération ; (b) on propose aux arbres des accointances de tomber dans la direction du plateau d'origine (même cheminement des messages que précédemment). Si l'arbre récupère un booléen à *true*, il envoie de nouveau les deux messages de propagation.

Effection et ses sous-classes sont mises en oeuvre par **act** >> "*self effectionArbres*" de EnvironmentForest.

Variables d'instance.

- *tabPropagationChablisSelonDiametre* : initialisé par **initialize** de la manière suivante :

1	(40 0)	Diamètre chablis primaire ≤ 40 cm : pas de chablis secondaire
2	(50 3)	Diamètre compris entre 40 et 50 cm : 3 chablis secondaires
3	(70 4)	Diamètre compris entre 50 et 70 cm : 4 chablis secondaires
4	(10000 4)	Diamètre ≥ 70 cm : 4 chablis secondaires.

2.4.10 - EffectionAngelique

Sous-classe de EffectionGrise, instanciée au moment de l'appel à la superclasse ModuleInstanceUnique (voir § 2.4.3). Elle renseigne la variable *moduleEffection* de EspecGrise. Elle possède bien évidemment toutes les propriétés de la superclasse précédente et elle gère la reproduction des Angéliques à travers la méthode **effectDissemination: moduleAgent**.

Si le destin de l'arbre est de disséminer des graines (champ n°4 de *destin*), EffectionAngelique active la méthode **disseminationGraines: libelleEspece effectif: <>** de Placeau pour successivement : (a) les placeaux proches, (b) les placeaux de la 2ème couronne, (c) les placeaux de la 3ème couronne sans les 12 placeaux des coins.

Dans le cas de l'Angélique, les effectifs disséminés (nombre de graines) sont les suivants :

		(12)	(12)	(12)		
	(50)	(50)	(50)	(50)	(50)	
(12)	(50)	(100)	(100)	(100)	(50)	(12)
(12)	(50)	(100)	(132)	(100)	(50)	(12)
(12)	(50)	(100)	(100)	(100)	(50)	(12)
	(50)	(50)	(50)	(50)	(50)	
		(12)	(12)	(12)		

2.4.11 - EffectioPetitGris

Sous-classe de EffectioGrise, instanciée au moment de l'appel à la superclasse ModuleInstanceUnique (voir § 2.4.3). Elle renseigne la variable *moduleEffectio* de EspeceGrise, lorsque l'arbre a été recruté au cycle précédent. L'affectation est faite par **initializeJeune** de EspeceGrise.

Un jeune recruté ne peut pas être chablis primaire (voir fonctionnement de DeliberationPetitGris) : il n'est donc pas concerné par les méthodes de propagation de sa superclasse.

Sa seule méthode spécifique d'effectio (**effectCroissance: moduleAgent de: croissance**) active la méthode correspondante de sa superclasse, puis indique à l'arbre *initializeVieux*, ce qui le "propulse" à l'âge adulte ...

2.5 - Les stades juvéniles

En l'état actuel du simulateur, on gère 4 stades de régénération (configuration adaptée au cas de l'Angélique) : le stade « graines », le stade « plantules » désignant les individus de moins de 1m50 de hauteur, le stade « gaules » désignant les individus faisant plus de 1m50 de hauteur et moins de 5 cm de diamètre et le stade « perches » qualifiant les jeunes arbres de diamètre compris entre 5 cm et 10 cm.

Au moment de l'initialisation des placeaux (EnvironmentForest → **initialize: cellsSize >> initializeCells >> Placeau** → **initialize: environmentForest**), la variable *stadeJuvenile* est renseignée de la manière suivante :

"stadeJuvenile:= SelvaOptions dicoEspècesEtInstancesClassesJuveniles". Cette méthode de Options va chercher toutes les clés d'entrée du dictionnaire *groupes* (variable de Options). Ces clés sont des libellés de groupes d'espèces fournis au départ par l'utilisateur. SelvaOptions cherche les modules de délibération associés à ces clés et leur envoie le message : **juvenileAssocie**, qui lui retourne le nom de la classe StadeJuvenile(Espece) associée à chaque module Deliberation(Espece). Alors SelvaOptions crée une instance de StadeJuvenile(Espece), l'initialise et la "range" dans la variable *stadeJuvenile* de Placeau à la clé #libelleEspece.

Ensuite, après avoir demandé aux arbres de délibérer ("*self* deliberationArbres"), la méthode **act** de EnvironmentForest lance le message : "*self* effectioArbresDisseminateurs". Cette méthode prend les arbres de la liste *listeArbresDisseminateurs* et leur demande : **dissemineGraines**, avant de remettre la liste à 0. Si l'arbre considéré est vivant, il s'adresse à son module d'effectio en lui demandant : "*effectDissemination: self*". Ce dernier avertit alors les placeaux voisins qu'ils vont recevoir des graines (méthode **disseminationGraines: (moduleAgent libelleEspece) effectif: effectifGraines** de Placeau : "*((stadeJuvenile at: libelleEspece) apportGraines: effectifGraines*", message qui a pour effet d'incrémenter la variable *apportGraines* de StadeJuvenile(Espece).

NB. Le booléen *stadeJuvenile* (variable de EnvironmentForest) entré par l'utilisateur au niveau de FenetreMain est testé par le module Deliberation(Espece) avant que celui-ci examine l'éventualité d'une floraison. Si ce booléen est à *true*, les placeaux disposent donc d'un stock de graines avant l'appel par **act** à "*self* deliberationPlaceaux".

EnvironmentForest → **act >> deliberationPlaceaux >> Placeau** → **choisiTonDestin: executerStadeJuvenile >> "stadeJuvenile do: [:uneEspece | arbresANaitre add: (uneEspece calculRecrutesEtMiseAJour: self dicoEvolution: (dicoCycle:= self initEvolution))]**, si le paramètre booléen *executerStadeJuvenile* est à *true*.

Dans cette dernière fonction, le paramètre passé *dicoCycle* est un nouveau dictionnaire dont seule la clé #surfaceTerrierePonderee a été renseignée. Il est destiné à être ajouté à *evolution* (variable de Placeau, voir sa description au § 2.3.2) dans la "colonne" (*self* owner cycleCourant). La fonction **calculRecrutesEtMiseAJour: placeau dicoEvolution: dicoCycle** va créer un nouveau dictionnaire consacré à l'espèce en cours d'étude et va remplir ses différents champs : #graines, #plantules, #gaules, #perches et #recrutes, avant (a) de l'ajouter à *dicoCycle* à la clé #espece et (b) de renvoyer un couple (code espèce, nombre de recrutés de plus de 10 cm de diamètre) à *arbresANaitre*.

Les différents stades de la régénération sont gérés par 4 classes : PopulationGraines, PopulationPlantules, PopulationGaules et PopulationPerches, toutes sous-classes de Population. Chaque stade gère ses propres règles de mortalité et de recrutement. La dynamique de passage entre les différents stades est assurée par la méthode ci-

dessus (**calculRecrutesEtMiseAJour: placeau dicoEvolution: dicoCycle**), qui fait appel à une fonction de Population (donc commune à tous) : **calculRecrutesEtMiseAJour: placeau recrues: apport**. Les effectifs fournis en paramètre servent d'abord à calculer un taux de passage. Ensuite, ils sont mis à jour une première fois par déduction du passage. Le chiffre obtenu sert à calculer un taux de mortalité, et les effectifs sont mis à jour une 2ème fois. L'effectif final résulte donc du calcul :

$$\text{effectifFinal} := (\text{effectifDepart} - \text{passage} - \text{morts} + \text{apport}),$$

avec passage = f(effectifDepart) et mort = g(effectifDepart - passage).

La fonction renvoie le paramètre *passage* à StadeJuvenile qui ré-appelle aussitôt la même fonction en lui passant ce paramètre, mais cette fois pour le compte de la sous-classe gérant le stade suivant de régénération.

NB. La variable *effectifs*, qui représente *effectifDepart* dans le calcul, est une variable d'instance de Population. Pour la même sous-classe, elle conserve sa valeur d'un cycle à l'autre, ce qui permet d'avoir "un stockage" pendant plusieurs cycles dans un stade donné, sauf pour PopulationGraines : en effet, la mortalité dans ce cas est de 100% d'un cycle à l'autre, si bien que seul l'apport en graines à chaque cycle alimente *effectifFinal*. Les graines n'ont que 2 devenir possibles : passer au stade supérieur ou mourir.

Cette variable *effectifs* est bien entendu propre à chaque sous-classe.

Lors du démarrage des simulations, *effectifs* vaut 0 pour toutes les sous-classes. Donc, le premier taux de passage graines → plantules vaut 0, (idem pour les autres stades). Au 2ème cycle, il y aura un passage et donc une alimentation de PopulationPlantules, mais le taux de passage plantules → gaules sera toujours à 0. Et ainsi de suite. Les différents stades se rempliront cycle après cycle : il faudra 4 cycles avant que la régénération devienne un phénomène "courant". Donc 12 ans au minimum pour qu'un arbre (ici une Angélique) parvienne au diamètre de recrutement (10 cm). Ceci est cohérent avec les observations de terrain (Rankin de Merona, comm. pers.).

A noter enfin que la fonction **dicoEspeciesEtInstancesClassesJuveniles** de SelvaOptions crée une instance de StadeJuvenile(Espece), et non de StadeJuvenile qui en est la superclasse. C'est StadeJuvenile(Espece) qui crée et initialise les instances de PopulationGraines, PopulationPlantules, PopulationGaules et PopulationPerches, pour son propre compte : elle les attribue aux différentes variables *graines*, *plantules*, *gaules* et *perches* de sa superclasse.

2.5.1 - StadeJuvenile

Sous-classe de Object, et superclasse de StadeJuvenile(Espece). Elle se charge d'assurer la dynamique de passage d'un stade de la régénération à l'autre, de renseigner le tableau *evolution* de Placeau (méthode **calculRecrutesEtMiseAJour: placeau dicoEvolution: dicoEvolution**), de permettre l'accès aux instances de PopulationGraines, PopulationPlantules, PopulationGaules et PopulationPerches stockées dans ses variables d'instance et de renvoyer à Placeau les couleurs associées à chaque stade (fonction **couleur: population**).

La méthode **evolutionEclaircie: dicoEvolution** permet de renseigner le tableau *evolution* de Placeau lorsque l'on réalise une éclaircie, c'est-à-dire que l'on introduit un cycle fictif : voir explication au § 2.4.1. EnvironmentForest → **eclaircie: critere** >> Placeau → **evolutionEclaircie** >> StadeJuvenile → **evolutionEclaircie: dicoCycle**. Cette méthode met tout simplement à -1 les champs #graines, #plantules, #gaules, #perches et #recrutes de toutes les espèces considérées.

Les méthodes de classe **listeCodesSousClasses** et **listeNomsSousClasses** sont appelées par la méthode de classe **menuEspece** de FenetrePlaceau (et de FenetreForest). Parmi les variables d'instance de FenetrePlaceau, figure *menuEspeceHolder* :

- *menuEspeceHolder* : stocke le nom de l'espèce pour laquelle existent des infos concernant la régénération. Par défaut, cette variable vaut "*#angelique asValue*". Tout changement provoque l'activation du message : **initializeStadesJuveniles.**"

La méthode **initializeStadesJuveniles** de FenetrePlaceau demande : "*stadeJuvenile:= self placeau libelleEspeceToStadeJuvenile: (menuEspeceHolder value)*". Cette dernière fonction vérifie la présence éventuelle d'une instance de StadeJuvenile(Espece) à la clé #libelleEspece (que retourne "*menuEspeceHolder value*") du tableau *stadeJuvenile* (méthode de Placeau). S'il n'y a rien, les champs

"graines", "plantules", "gaules" et "perches" de FenetrePlaceau sont mis à 0. S'il existe une instance de StadeJuvenile(Espece), les variables de StadeJuvenile renvoient l'info *effectifs* de leur superclasse Population.

Variables d'instance.

- *apportGraines* : initialisée à 0 au moment de l'instanciation des sous-classes StadeJuvenile(Espece). Mise à jour par Placeau au moment de l'ordre EF → **act** >> **effectifArbresDisseminateurs** (voir explications en début de chapitre). Cette variable alimente la variable *effectifs* de PopulationGraines. Elle est remise à 0 avant le passage au cycle suivant.

- *gaules* : contient l'instance de PopulationGaules créée par chaque sous-classe StadeJuvenile(Espece).

- *graines* : contient l'instance de PopulationGraines créée par chaque sous-classe StadeJuvenile(Espece).

- *perches* : contient l'instance de PopulationPerches créée par chaque sous-classe StadeJuvenile(Espece).

- *plantules* : contient l'instance de PopulationPlantules créée par chaque sous-classe StadeJuvenile(Espece).

2.5.2 - StadeJuvenileAngelique

Sous-classe de StadeJuvenile. Une instance de cette classe est créée par SelvaOptions, au moment de l'initialisation de chaque placeau. Elle-même, au moment de son initialisation, crée une instance de chacune des 4 classes : PopulationGraines, PopulationPlantules, PopulationGaules et PopulationPerches qu'elle stocke dans les variables correspondantes de sa superclasse.

Les autres méthodes fournissent uniquement des informations (code espèce, libellé espèce, etc.).

2.5.3 - Population

Sous-classe de Object et superclasse d'un grand nombre de choses, comme exposé dans le tableau ci-après.

<u>Population</u>
<p><u>PopulationEffectifsNumeriques</u> Met à jour les effectifs de façon simple.</p> <ul style="list-style-type: none"> • <u>PopulationEffectifsCollectifs</u> Transforme en entiers tous les nombres transmis, de manière à en faire de vrais effectifs. <ul style="list-style-type: none"> → <u>PopulationGraines</u> → <u>PopulationPlantules</u> <u>PopulationEffectifsIndividuels</u> • Réalise des tirages au hasard pour décider de la mort ou du passage des individus "gaules" et "perches". Renvoie également un intervalle de couleurs unique pour la représentation graphique de ces éléments. <ul style="list-style-type: none"> → <u>PopulationGaules</u> → <u>PopulationPerches</u> <p><u>PopulationEffectifsBooleens</u> Cette classe n'est pas utilisée actuellement. Elle est destinée à gérer des booléens : présence/absence d'un stade de la régénération.</p>

La méthode **calculRecrutesEtMiseAJour: placeau recrues: apport**, appelée à 4 reprises par StadeJuvenile (une fois pour chaque sous-classe finale) met à jour le stock d'individus présent sur le placeau et décide de l'effectif qui va passer au stade suivant. Elle fait pour cela appel aux méthodes **tauxDePassage: placeau** et **tauxDeMort: placeau** de ses sous-classes, ainsi qu'à la fonction **miseAJourEffectifs**. **NB.** Cette dernière fonction est définie à deux niveaux : PopulationEffectifsNumeriques (mise à jour directe de *effectifs*, pour les gaules et les perches) et PopulationEffectifsCollectifs (mise à jour après test de *effectifs*, pour

les graines et les plantules : en effet, les différents calculs pour ces deux stades sont arrondis ce qui peut provoquer l'apparition d'effectifs négatifs).

La méthode **couleur** recherche dans SelvaOptions la couleur correspondant à un niveau donné d'intensité de présence pour les différents stades de la régénération.

Variables d'instance.

- *effectifs* : initialisée à 0 au moment de l'instanciation des différentes sous-classes PopulationGraines, PopulationPlantules, PopulationGaules et PopulationPerches. Pour chacune d'elles et d'un cycle à l'autre, cette variable conserve sa valeur (remise à jour après déduction du passage dans la classe supérieure, de la mortalité des individus restants et des apports) ce qui permet une accumulation éventuelle, sauf pour les graines.

2.5.4 - PopulationEffectifsNumeriques

Sous-classe de Population et superclasse de PopulationEffectifsCollectifs (voir tableau ci-dessus). Une seule méthode : mise à jour simple de la variable *effectifs* : méthode **miseAJourEffectifs: eff**.

2.5.5 - PopulationEffectifsCollectifs

Sous-classe de PopulationEffectifsNumeriques et superclasse de PopulationGraines et PopulationPlantules. La mise à jour des effectifs se fait après test d'une valeur éventuellement négative. Une fonction importante est : **arrondiPondere: nb**, qui transforme *nb* en entier en l'arrondissant au hasard à la valeur supérieure ou à la valeur inférieure. Indispensable car les deux stades de régénération concernés se voient appliquer des formules de calcul (taux de passage et taux de mortalité) portant sur des effectifs globaux et non sur des individus (trop nombreux !).

2.5.6 - PopulationEffectifsIndividuels

Sous-classe de PopulationEffectifsNumeriques et superclasse de PopulationGaules et PopulationPerches.

Les méthodes **tauxDeMort: placeau** et **tauxDePassage: placeau** font appel, pour le calcul, à (a) une probabilité (respectivement de mort et de passage dans la classe supérieure) propre à chaque sous-classe et (b) une fonction de tirage de nombres aléatoires : **proba: p nbTirages: nb** à laquelle les probabilités précédemment évoquées et la variable *effectifs* sont passées en paramètres.

La méthode **proba: p nbTirages: nb** tire des nombres aléatoires autant de fois qu'il y a d'individus dans *effectifs*, et elle les compare à la valeur de la probabilité fournie.

Un tableau de couleurs par défaut est fourni pour la représentation graphique (appelé par la fonction **couleur de Population**), valable à la fois pour les gaules et pour les perches.

2.5.7 - PopulationGraines

Sous-classe de PopulationEffectifsCollectifs. Une instance est créée par StadeJuvenile(Espece), au moment de son initialisation par SelvaOptions.

La fonction **tauxDeMort: placeau** renvoie la variable *effectifs*, si bien qu'il n'y a jamais accumulation de graines d'un cycle à l'autre : soit elles deviennent plantules, soit elles meurent. La fonction **tauxDePassage: placeau** renvoie 10% des effectifs de départ.

Un tableau de couleurs par défaut est fourni pour la représentation graphique (appelé par la fonction **couleur de Population**).

2.5.8 - PopulationPlantules

Sous-classe de PopulationEffectifsCollectifs. Une instance est créée par StadeJuvenile(Espece), au moment de son initialisation par SelvaOptions.

La fonction **tauxDeMort: placeau** renvoie la variable *effectifs* multipliée par 2/3. La fonction **tauxDePassage: placeau** renvoie la quantité :

"*^self arrondiPondere: [((stpm - (placeau surfaceTerrierePonderee)) / (placeau environnement surfaceTerrierePondereeMaxi)) / (3 * effectifs)]*". En fait, on calcule la place théorique restant sur le plateau et le nombre de plantules pouvant devenir des gaules est proportionnel à la place disponible. Cette place disponible est évaluée comme étant la différence entre l'indice *surfaceTerrierePonderee* sur le plateau et la valeur maxi. atteinte par cette variable sur l'ensemble des plateaux de l'univers d'étude.

Un tableau de couleurs par défaut est fourni pour la représentation graphique (appelé par la fonction **couleur de Population**).

2.5.9 - PopulationGaules

Sous-classe de PopulationEffectifsIndividuels. Une instance est créée par StadeJuvenile(Espece), au moment de son initialisation par SelvaOptions.

Elle a pour rôle essentiel de fournir à sa superclasse deux probabilités : une de mort et une de passage dans la classe supérieure. Elles les calcule de la manière suivante :

probaDeMort

0,6

probaDeRecrutement: placeau

| x |

$x := e^{((7 * \text{placeau surfaceTerrierePonderee}) + 1)}$

$^1 - (x / (1+x))$.

2.5.10 - PopulationPerches

Sous-classe de PopulationEffectifsIndividuels. Une instance est créée par StadeJuvenile(Espece), au moment de son initialisation par SelvaOptions.

Elle a pour rôle essentiel de fournir à sa superclasse deux probabilités : une de mort et une de passage dans la classe supérieure. Elles les calcule de la manière suivante :

probaDeMort

0,333

probaDeRecrutement: placeau

| x |

$x := e^{(10 * \text{placeau surfaceTerrierePonderee})}$

$^1 - (x / (1+x))$.

2.6 - Les "outils" de SELVA

2.6.1 - Options

Sous-classe de Object. Cette classe gère 2 dictionnaires importants de SELVA : *groupes* et *couleurs*. Une instance est créée lors du cliquage sur l'icône "*Selva*" de ForestLauncher, de la manière suivante (méthode de classe **makeGlobalVariable**) :

"*SelvaOptions isNil ifFalse: [^SelvaOptions]*."

^SelvaOptions := self new initialize."

Donc, la création n'est pas systématique. En pratique, lorsqu'on quitte une session de travail de VisualWorks 1.0, SelvaOptions n'est pas mise à nil et on la retrouve à la session suivante. Le seul cas dans lequel la méthode **initialize** est appliquée, c'est lors du premier chargement de SELVA (méthode **chargePremiereFois** de Tools). **MAIS** : il y a moyen de réinitialiser Options par le biais du menu général de ForestLauncher : en effet, la méthode **simulationMenu** propose à l'utilisateur une étiquette "*Reinitialisation SelvaOptions*" qui envoie le message : "*Options initializeSelvaOptions*". Celle-ci a pour effet de créer une nouvelle instance de Options.

La méthode **initialize** provoque la mise à jour par défaut des dicos *groupes* (**initGroupes**) et *couleurs* (**initCouleurs**). Par la suite, ces dicos sont modifiés et sauvegardés à travers deux fenêtres de saisie : FenetreOptionsCouleurs pour *couleurs* et FenetreEspeces pour *groupes*, qui sont les deux "donneurs d'ordre" essentiels.

Quelques méthodes importantes :

- **colorsCopy** : appelée par FenetreOptionsCouleurs (dans sa méthode **initialize**). Duplique les infos détenues par *couleurs* pour la mise à jour de ce dictionnaire.
- **couleurs: newColor** : appelée par FenetreOptionsCouleurs (méthode **valider**). Cette classe transmet à Options un nouveau dictionnaire contenant l'ancien *couleurs* mis à jour par l'utilisateur. La méthode commence par mettre à jour les champs #couleur de *groupes*, puis écrase l'ancien *couleurs* avec *newColor*.
- **groupCopy** : appelée par FenetreEspeces (méthode **initLibelles**). Duplique les infos détenues par *groupes* pour la mise à jour de ce dictionnaire.
- **groupes: newGroups** : appelée par FenetreEspeces (méthode **valider**). Cette classe transmet à Options un nouveau dictionnaire contenant l'ancien *groupes* mis à jour par l'utilisateur. La méthode commence par mettre à jour *couleurs*, puis écrase l'ancien *groupes* avec *newGroups*.
- **dicoEspecesEtInstancesClassesJuveniles** : méthode très importante, appelée par Placeau (**initialize: environnementForest**). Elle a pour effet de créer autant d'instances de StadeJuvenile(Espece) qu'il y a de méthodes **juvenileAssocie** dans les classes Deliberation(Espece) attachées aux clés d'entrée de *groupes*. Elle retourne à Placeau un dictionnaire indexant ces instances aux clés d'entrée.
- **defaultEspece** (méthode de classe) : appelée par FenetreEspeces (méthode **ajoutLibelle**). Lorsque l'on entre un nouveau nom de groupe, ce nom est récupéré "*asSymbol*" et devient une clé d'entrée de *groupes*. A cette clé est ajoutée un dico. par défaut grâce à **defaultEspece**, en attendant que ses champs soient renseignés par l'utilisateur (toujours à travers la fenêtre FenetreEspeces). Le travail se terminera par la méthode **valider** qui fera appel à **groupes: newGroups** (voir ci-dessus).
- **save** : sauvegarde l'instance d'Options dans un fichier BOSS. Appelée au niveau du menu général de ForestLauncher (méthode **optionsMenu**).
- **loadSelvaOptions** (méthode de classe) : symétrique de la méthode précédente, récupère l'instance d'Options à partir du fichier "*SelvaOptions.boss*". Elle est également appelée par le menu de ForestLauncher (méthode **optionsMenu**).

Variables d'instance

- *couleurs* : dico. initialisé après *groupes* par la méthode **initCouleurs**. Après cette initialisation, il a l'allure suivante :

Caractéristiques des groupes	Valeur par défaut des variables
#fond	ColorValue lightGray
#graines	nuances2: (ColorValue white)
#plantules	nuances2: (ColorValue white)
#gaules	nuances2: (ColorValue white)
#perches	nuances2: (ColorValue white)
#selection	ColorValue black
#chablisPrimaire	ColorValue blue
#chablisSecondaire	ColorValue white
#mortSurPied	ColorValue yellow
#mortParEclaircie	ColorValue black
#especeGrise	ColorValue darkGreen
#especeRecrute	ColorValue green
#angelique	ColorValue red

Les champs correspondant aux espèces et groupes d'espèces sont remplis à partir du champ #couleur qui leur est associé dans le dico. *groupes*.

- *groupes* : dico. initialisé par la méthode **initGroupes**. Après cette initialisation, il a l'allure suivante :

Noms des espèces ou groupes d'espèces étudiés	Caractéristiques des groupes	Valeur par défaut des variables
#especeGrise	#diametreElim	40
	#randomAutocorrele	<i>true</i>
	#codeEssence	OrderedCollection new
		-
		-
		-
#especeRecrute	#classe	#DeliberationGrise
	#couleur	ColorValue darkGreen
	#diametreElim	40
	#randomAutocorrele	<i>true</i>
	#codeEssence	100
#angelique	#classe	#DeliberationGrise
	#couleur	ColorValue green
	#diametreElim	40
	#randomAutocorrele	<i>true</i>
	#codeEssence	103
	#classe	#DeliberationAngelique
	#couleur	ColorValue red

2.6.2 - Tools

Sous-classe de Object, cette classe gère un certain nombre d'utilitaires d'intérêt général. Les principaux sont proposés par le menu général de ForestLauncher via la méthode **simulationMenu** :

- "*Tools new memControl: true*" : nettoie la mémoire, puis collecte toutes les instances encore présentes dans le système, ainsi que les process encore actifs. Elle affiche leur effectif respectif dans la fenêtre Transcript, et inspecte la liste des instances (le paramètre étant à *true*).

- "**Tools new memClear**" : nettoie la mémoire, termine tous les process en cours et détruit les instances encore existantes.

- "**Tools new saveForest**" : sauvegarde les différentes classes de SELVA par catégories en utilisant par défaut le nom de ces catégories et place les fichiers dans un répertoire spécifié par l'utilisateur.

- "**Tools new chargeSelva**" : demande le nom du répertoire contenant les ".save" et charge les fichiers suivants :

- Object-SELVA.save,
- SmallInteger-SELVA.save,
- ByteString-SELVA.save,
- SELVA-Tools.save,
- SELVA-Windows.save,
- SELVA-MVC.save,
- SELVA-Simulation.save,
- SELVA-Arbres.save,
- SELVA-Juvenile.save.

NB. Dans ce cas, il n'y a pas réinitialisation de SelvaOptions.

Les autres méthodes importantes sont :

- **chargePremiereFois**, qui permet de reconstruire SELVA à partir d'une image vierge. **NB** : à ce niveau, la déclaration des variables globales *RandomUnif* et *Pi* est faite, et SelvaOptions est initialisée. Pour utiliser cette méthode, il faut l'avoir sauvegardée dans un fichier (`chargeSELVA.save`, par exemple) sur lequel on utilise un "file editor" (voir Utilities de SmallTalk). On évalue ensuite le texte par "*do it*".

- **quiComprend: methode** : permet de trouver quelle classe est capable de comprendre la méthode spécifiée et donc à qui elle est destinée.

- **save: object onFileWithConfirm: aString** (méthode de classe) : sauvegarde l'objet spécifié dans un fichier BOSS dont le nom est fourni par l'utilisateur.

- **loadWithConfirm: aString** (méthode de classe) : symétrique de la méthode précédente. Charge le fichier BOSS dont le nom est fourni par l'utilisateur.

2.6.3 - WriteFile

Sous-classe de Object. Une instance est créée par EnvironmentForest dans **sortieFichierFic: confirmation complet: boolean** et **sortieFichierStadeJuvenile: confirmation**.

Cette classe s'occupe d'ouvrir un fichier vierge, de gérer l'écriture d'infos sur ce fichier et de le fermer.

Variables d'instance.

- *file* : pointeur sur le fichier ouvert en écriture.

- *name* : nom du fichier créé.

- *oldCursor* : stocke l'apparence **currentCursor** du curseur qui affiche un logo d'écriture pendant que le fichier est ouvert.

2.6.4 - FileAsArray

Sous-classe de Object. Une instance est créée par FenetreMain dans **initEnvironmentFic: nomFichier**. Cette classe ouvre un fichier contenant des informations qu'elle stocke dans un tableau de type OrderedCollection.
Ex : "*param:= FileAsArray new readTab: (nomFichier, '.stat')*". *param* devient un tableau à 2 champs contenant notamment la dimension de l'univers en placeaux ("*param at:1*") et le numéro de la campagne de démarrage ("*param at: 2*").

La fonction **readWord** se charge de récupérer dans le fichier tous les caractères consécutifs qui ne sont ni des blancs, ni des <CR>, ni des <LF>. Dès qu'elle rencontre l'un de ces séparateurs, elle renvoie le mot qu'elle a constitué à l'appelant (**readLine**) qui le stocke dans un champ du tableau résultat.

Variables d'instance.

- *file* : pointeur sur le fichier ouvert en lecture.

- *name* : nom du fichier d'entrée.

2.6.5 - **ForestRandomUnif**

Sous-classe de Random. Une instance de cette classe est créée par EnvironmentForest (dans initializeSysteme), et stockée dans RandomUnif, variable globale déclarée lors du premier chargement de SELVA (voir Tools : "SmallTalk declare: #RandomUnif from: ForestRandomUnif.").

Dans Random, "*self next*" renvoie une variable aléatoire uniformément distribuée sur l'intervalle [0,1]. La fonction nextInteger: max de ForestRandomUnif renvoie un nombre entier tiré dans l'intervalle [0,max], selon une loi uniforme.

2.6.6 - **Gauss**

Sous-classe de Object et superclasse de Gauss0, Gauss1 et GaussMaxMuller. Utilisée en phase de programmation par Caruso. Elle permet de générer la courbe de la loi normale.

Elle gère également une sortie fichier : sortieFichier: n nom: nomFichier permet d'écrire dans *nomFichier* les n premiers résultats du tirage aléatoire.

Variables d'instance.

- *mu* : moyenne de la distribution normale. Initialisée à 0.

- *ran* : instance de Random (affectée par la fonction initialize).

- *sigma* : écart-type de la distribution. Initialisé à $\sqrt{0,2}$.

2.6.7 - **Gauss1**

Sous-classe de Gauss. Une instance est créée par Deliberation pour intervenir dans les calculs de croissance. Elle redéfinit la fonction next de Random en réalisant la sommation de 12 variables aléatoires uniformément distribuées.

NB. GaussMaxMuller gère une autre méthode de génération d'une loi normale à partir de lois uniformes.

2.6.8 - **ForestTime**

Sous-classe de Object. Compte le temps nécessaire à la réalisation de différentes opérations (utilisée en phase de programmation).