



Modélisation de la compétition après jardinage avec dégagement accru des tiges élités

Projet de recherche réalisé dans le cadre du projet « Mise en place d'un aménagement à valeur ajoutée à l'échelle de l'arbre des feuillus tolérants à l'ombre de l'est du Canada : Modélisation de la faisabilité et essai pilote sur un territoire au Nouveau-Brunswick »

(Projet Swift F09 du Centre Canadien sur la Fibre de Bois)

Rapport final

Préparé par

Frédéric Doyon, ing. f., Ph.D.

Remis à Edwin Swift, Service Canadien des Forêts, Ress. Nat. Canada

Mars 2010

Tables des matières

Liste des Figures	2
Liste des Tableaux	4
Liste des Annexes	4
Remerciements.....	5
Remerciements.....	5
Citation.....	5
Résumé	6
1. Introduction.....	7
Objectifs de recherche.....	10
Approche analytique.....	10
Hypothèses	10
2-Méthodologie.....	12
Jeu de données.....	12
Calcul de l'indice de compétition	2
Modélisation du prélèvement balancé	3
Modélisation du prélèvement avec dégagement accru des tiges élites.....	4
Simulations	6
Analyses	6
3 – Résultats.....	8
Description du jeu de données.....	8
Moyenne générale des différences de changement de compétition après traitement	10
Type de prélèvement	11
Effets simples sur l'indice de différence de changement de compétition après traitement	11
Modèle prédictif de l'indice de différence de changement de compétition après traitement	17
4 – Discussion	21
Modélisation des effets simples du changement de la différence de compétition après traitement.....	21
La modélisation de l'effet du dégagement accru sur la croissance dans le modèle <i>Cohorte</i>	23
Évaluation de l'effet du dégagement accru sur la croissance dans le modèle <i>Cohorte</i> ...	23
5 – Conclusion.....	24
6 – Littérature citée	26

Liste des Figures

- Figure 1. Photo aérienne infrarouge d'un secteur de coupe venant d'être traité par jardinage par pied d'arbre. On peut facilement y observer la récurrence systématique des ouvertures. 10
- Figure 2. Les placettes échantillons ayant servis pour ce projet sont dans l'Outaouais, dans la partie ouest du domaine bioclimacique de l'érablière à bouleau jaune. 12
- Figure 3. Création d'une zone tampon autour de la parcelle-échantillon pour éviter l'effet de bordure lors du calcul des indices de compétition. Les zones avec un nombre primé reproduisent la zone avec le même nombre à l'intérieur de la PÉ..... 2
- Figure 4. Répartition spatiale des arbres avant et après traitement dans une des PÉs traitées par a) CJ et par b) CJ+ par l'algorithme de récolte partielle utilisé dans cette étude. Les arbres sont représentés par des cercles. avec le contour en bleu sont les arbres résiduels après traitement, les arbres en rouge avec un « X » à l'intérieur sont les arbres prélevés et les arbres en vert plein sont les tiges élites. Le prélèvement est de 30%. La surface terrière en tiges élites dans la CJ+ est de $1\text{m}^2/\text{ha}$ 6
- Figure 5. Relation entre la moyenne quadratique du DHP et la surface terrière des PÉs. 10
- Figure 6. Différence de la moyenne quadratique avant et après traitement pour la CJ et la CJ+. 11
- Figure 7. Relation entre le DHP de la tige et l'indice de la différence de changement de compétition après traitement pour les tiges élites. 13
- Figure 8. Relation entre le DHP de la tige et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage..... 13
- Figure 9. Relation entre la surface terrière initiale et l'indice de la différence de changement de compétition après traitement pour les tiges élites. 14
- Figure 10. Relation entre la surface terrière initiale et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage. 14
- Figure 11. Relation entre le taux de prélèvement en surface terrière et l'indice de la différence de changement de compétition après traitement pour les tiges élites. 14

Figure 12.	Relation entre le taux de prélèvement en surface terrière et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage. ..	15
Figure 13.	Relation entre la surface terrière en tiges élites et l'indice de la différence de changement de compétition après traitement pour les tiges élites.	16
Figure 14.	Relation entre la surface terrière en tiges élites et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.	16
Figure 15.	Relation entre la différence de la moyenne quadratique de DHP après traitement et l'indice de la différence de changement de compétition après traitement pour les tiges élites.	17
Figure 16.	Relation entre la différence de la moyenne quadratique de DHP après traitement et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.	17
Figure 17.	Croissance dans la version paramétrisée pour le Nouveau-Brunswick de <i>Cohorte</i> pour des érables à sucre identifiés comme tiges élites en fonction de l'indice de compétition subie par la tige après CJ et CJ+ pour les tiges vigoureuses et non-vigoureuses.	24

Liste des Tableaux

Tableau 1.	Composition et structure des placettes échantillons ayant servis pour ce projet.	8
Tableau 2.	Tableau de l'analyse de variance de la régression linéaire entre les variables du modèle et la différence de changement de compétition après traitement. .	12
Tableau 3.	Étape de la sélection pas-à-pas des termes du modèle de régression multiple pour les tiges élites	18
Tableau 4.	Coéfficients des termes utilisés dans le modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges élites	18
Tableau 5.	Analyse de variance du modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges élites. PRELEVEMEN=Prélèvement, PW_MARTELA= St élites, ST_INIT=St intital.....	19
Tableau 6.	Étape de la sélection pas-à-pas des termes du modèle de régression multiple pour les tiges de bourrage.....	20
Tableau 7.	Coéfficients des termes utilisés dans le modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges de bourrage	20
Tableau 8.	Analyse de variance du modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges de bourrage. PRELEVEMEN=Prélèvement, PW_MARTELA= St élites, ST_INIT=St intital.	21

Liste des Annexes

Annexe 1.	Code source Visual Basic pour le prélèvement balancé et avec dégagement accru des tiges élites.....	29
-----------	---	----

Remerciements

J'aimerais en tout premier lieu remercier Pascal Rochon de l'Institut québécois d'Aménagement de la Forêt feuillue (IQAFF) pour son magnifique travail de programmation et d'analyse. Ce projet est une collaboration fructueuse avec Martin Béland de la Faculté de Foresterie de l'Université de Moncton, Campus d'Edmundston, et Edwin Swift, du Service Canadien des Forêts, Ressources Naturelles Canada. Leur contribution m'a été précieuse dans l'avancement de mes travaux et je leur en remercie. Je souligne l'accueil chaleureux qui m'a été accordé lors de mon séjour parmi eux. Je remercie le Centre Canadien sur la Fibre de Bois pour le financement de ma participation dans ce projet. Il faut assurément souligner l'apport très important des différentes sources de données pour les grandes parcelles-échantillons qui ont été utilisées dans ce projet, dont celles provenant de la MRC des Collines; merci à Vincent Barrette avoir en donné l'autorisation. Finalement, je remercie Régis Pouliot pour sa contribution essentielle dans la programmation de *Cohorte* et pour son soutien lors des simulations faites à l'IQAFF et à l'Université de Moncton, Campus Edmundston.

Citation

Doyon, F. 2010. Modélisation de la compétition après jardinage avec dégagement accru des tiges élites. Rapport technique de l'Institut québécois d'Aménagement de la Forêt feuillue. Réalisé dans le cadre du projet « Mise en place d'un aménagement à valeur ajoutée à l'échelle de l'arbre des feuillus tolérants à l'ombre de l'est du Canada : Modélisation de la faisabilité et essai pilote sur un territoire au Nouveau-Brunswick) (Projet Swift F09). Remis à Edwin Swift, du Centre canadien sur la Fibre du Bois. 28 p. + Annexe.

Résumé

L'aménagement de la forêt feuillue fait face à une grave crise, remettant en cause l'utilisation du jardinage par pied d'arbre tel qu'exécutée durant les 25 dernières années. Une approche alternative à ce traitement viserait à dégager explicitement les tiges d'avenir promettant le meilleur gain en valeur par un détournage partiel de la couronne. Un dispositif expérimental a été mis en place au Nouveau-Brunswick pour venir tester cette approche. Cependant, comme plusieurs années seront nécessaires avant d'obtenir les résultats de ce dispositif, nous aimerions pouvoir prédire les effets du jardinage avec dégagement accru des tiges élites (CJ+) sur le rendement des peuplements à l'aide de la modélisation. Dans cette étude, nous avons évalué comment varie la compétition autour des arbres élites et de bourrage après application de la CJ+ en comparaison avec celle qui serait obtenue sous un jardinage traditionnel (CJ). Pour ce faire nous avons utilisé 49 grandes parcelles-échantillons (PÉ) (~ 0.25 ha) de forêts feuillues provenant du domaine biocimacique de l'érablière à bouleau jaune et de l'érablière à tilleul de l'Ouest du Québec dans lesquelles tous les arbres ont été cartographiés. Puis nous avons appliqué un algorithme de récolte partielle visant à reproduire la CJ et la CJ+, sous différents niveaux de prélèvement (20-40%) et de surface terrière en tiges élites (0.5 à 9 m²/ha). Pour chaque traitement, 100 simulations ont été produites. Nous avons par la suite compilé les changements dans la compétition autour de chaque arbre après traitement à l'aide d'un indice de compétition qui tient compte de la position spatiale des arbres. Nos résultats montrent que le changement dans la différence de compétition après traitement est surtout fonction de la surface terrière en tiges élites à dégager; plus il y a de tiges élites, moins l'effet du dégagement accru sur celles-ci se fait sentir. La surface terrière initiale et le prélèvement ajoute un peu au pouvoir prédictif du modèle; plus celles-ci sont importantes, ainsi va l'effet du dégagement accru. Cependant, cet effet sur la croissance des tiges élites demeure modeste. Le dégagement accru des tiges élites ne semble pas nuire significativement aux tiges de bourrage par une augmentation sensible de la compétition. Nos résultats suggèrent donc qu'il est important de limiter le nombre de tiges élites dans l'application de la CJ+ sinon l'avantage de la sélection sur les tiges d'avenir est réduit considérablement. Tout cela nous indique qu'en régime de coupe partielle, la latitude pour faire une différence sur la croissance des tiges élites n'est pas aussi grande que nous l'avions présumée au départ de cette recherche. Le dispositif expérimental dans le cadre de ce projet visera justement à vérifier si l'application pratique ne se traduira pas en une hausse plus marquée de la croissance des tiges élites.

1. Introduction

La forêt feuillue tempérée occupe une part importante du territoire forestier canadien, couvrant le sud de l'Ontario jusqu'à la Nouvelle-Écosse. Les espèces d'arbres qui la composent peuvent générer une gamme diversifiée de produits forestiers. Or ces produits offrent un rendement en valeur ajoutée très différent selon la qualité de la tige et du bois. Conséquemment, la notion de qualité (bille et bois) est une composante importante des objectifs de production dans la foresterie de cette forêt. Cependant, tel que constaté lors des travaux de la Commission d'étude sur la gestion de la forêt publique québécoise (Commission Coulombe), la forêt feuillue tempérée a fortement été dégradée en qualité durant le dernier siècle suite à une succession de coupes partielles écrémant le capital bois d'oeuvre.

Depuis le début des années 1990, le ministère des ressources naturelles et de la faune (MRNF) du Québec, soucieux d'améliorer la production forestière et de corriger l'état d'appauvrissement des forêts, a fait la promotion de la coupe jardinage en se basant sur les résultats des expériences de Zoran Majcen (Majcen et al. 2005). D'après Majcen, cette méthode devait permettre de mieux de conserver les caractéristiques de la composition et de la structure des forêts tout en assurant un rendement de croissance et de régénération soutenu et propice au maintien de l'industrie forestière (Majcen 1990). Ainsi, Bédard et al. (2003) ont évalué qu'après 10 ans, l'accroissement de la surface terrière dans les parcelles jardinées s'élevait à 0.35 m²/ha/an comparativement à 0.14 m²/ha/an dans les parcelles non-traitées. Cependant, après près de 20 ans d'application sur les terres publiques par les compagnies forestières, les études montrent que les rendements visés au départ par la coupe jardinage ne sont pas atteints de façon opérationnelle, et même que certains peuplements et essences sont au contraire dégradés (Bédard et Brassard 2002, Bédard et al. 2004). Ces derniers auteurs imputent ces résultats à une mauvaise sélection des tiges lors du martelage, menant à un taux de mortalité supérieur à ceux obtenus dans les dispositifs de la Direction de la Recherche forestière du MRNF. Les rendements de croissance et de qualité espérés étant très faibles, il fut alors proposer de rallonger la rotation entre les différentes coupes.

Cette situation s'aggrava en 2006 avec la crise forestière liée au domaine du bois d'oeuvre et de la pâte. Sous les nouvelles contraintes de rendement et de priorité de récolte, la disponibilité de capital bois d'oeuvre de qualité en forêt est devenue très faible; les compagnies forestières peinaient à rentabiliser la coupe jardinage sans récolter les quelques

tiges de valeur restantes. De plus, les surplus de bois récoltés et destinés à l'industrie de la pâte étaient difficiles à écouler, dans le contexte économique où il y avait de moins en moins de preneurs de bois de trituration. La coupe de jardinage devenait alors difficilement praticable puisqu'il ne reste que très peu de marge de rentabilité pour exercer les pratiques permettant la restauration du capital bois d'œuvre.

Face à ces révélations, nous devons revoir l'utilisation systématique de la coupe jardinage dans les érablières du Québec et tenter de trouver des approches alternatives qui montrent des rendements intéressants tant en volume qu'en qualité et qui soient rentables (Lessard et al. 2005). Dans une de ses recommandations, la Commission Coulombe propose de mettre en place des mesures explicites de restauration de la qualité des bois de cette forêt. Il est urgent de développer une sylviculture axée sur la qualité du bois afin de maximiser la valeur des produits obtenus en fonction des investissements sylvicoles. Or, curieusement, ce n'est qu'assez récemment que les sylviculteurs se sont intéressés à la démonstration de l'effet de la sylviculture sur la qualité des tiges et du bois des feuillus durs dans un contexte de peuplements naturels.

Ce projet vise à développer une approche sylvicole axée sur la production de valeur ajoutée pour les feuillus tolérants à l'ombre de l'est du Canada. L'approche sylvicole visée par ce projet préconise la croissance rapide des tiges élites du peuplement par un détournement de cime lors des opérations de jardinage. Forget et al. (2007) ont montré comment la croissance des arbres après jardinage était dépendant du niveau de dégagement de la cime. Cependant, une telle approche sylvicole n'a pas été formalisée par le passé car on ne trouve pas dans la littérature de données empiriques présentant les rendements d'un jardinage avec dégagement accru des tiges élites. Conséquemment, ce projet vise donc à évaluer les rendements en valeur (croissance et qualité) que pourrait procurer cette approche sylvicole.

Afin de bien évaluer les rendements qui découleraient d'une approche sylvicole basée sur le détournement des cimes des tiges élites (ci-après nommé coupe de jardinage avec martelage positif ou CJ+) deux approches complémentaires ont été mises de l'avant : 1) un dispositif expérimental visant à obtenir des données empiriques sur les activités de planification et d'opérations du traitement et sur la réponse en rendement du peuplement résiduel et 2) l'utilisation d'un modèle de simulation qui permettrait de capturer l'effet du dégagement accru sur les tiges élites lors de l'évaluation de la croissance du peuplement. Ce rapport de recherche présente les accomplissements réalisés jusqu'à présent concernant la seconde approche utilisant la modélisation.

Le modèle utilisé pour évaluer l'effet du dégagement accru sur la croissance est *Cohorte* (Doyon et al. 2005 et 2006). *Cohorte* est un modèle de simulation de croissance par arbre individuel indépendant de la distance. Dans *Cohorte*, la croissance d'une tige est dépendante de son espèce, de sa vigueur et de sa position sociale, telle qu'exprimée par le rang que le diamètre de la tige occupe par rapport à l'ensemble des autres tiges du peuplement. Ainsi, sans toutefois prendre en considération l'organisation spatiale des tiges de la communauté des arbres, *Cohorte* estime un niveau général de compétition que la tige subit de ses congénères en fonction de son rang social. Cela suppose que ce rang est représentatif de la compétition subit par la tige en moyenne sur l'ensemble de l'unité de simulation (un hectare en occurrence dans *Cohorte*). Cette prémisse peut être satisfaisante pour exprimer l'état de la compétition après un jardinage par pied d'arbre occasionnant une répartition plutôt systématique des ouvertures dans le couvert (Figure 1). Or, la CJ+ vise justement à occasionner des ouvertures plus regroupées autour des arbres élites pour les dégager. Ainsi, cette prémisse ne peut plus être considérée comme valable dans l'évaluation de la compétition des tiges. En effet, pour un niveau de prélèvement donné, dans le cas de la CJ+, les tiges visant à être dégagées auront plus de chance d'avoir une compétition moindre qu'après une CJ alors qu'au contraire, les tiges faisant partie du bourrage auront plus de chance de subir une compétition plus élevée.

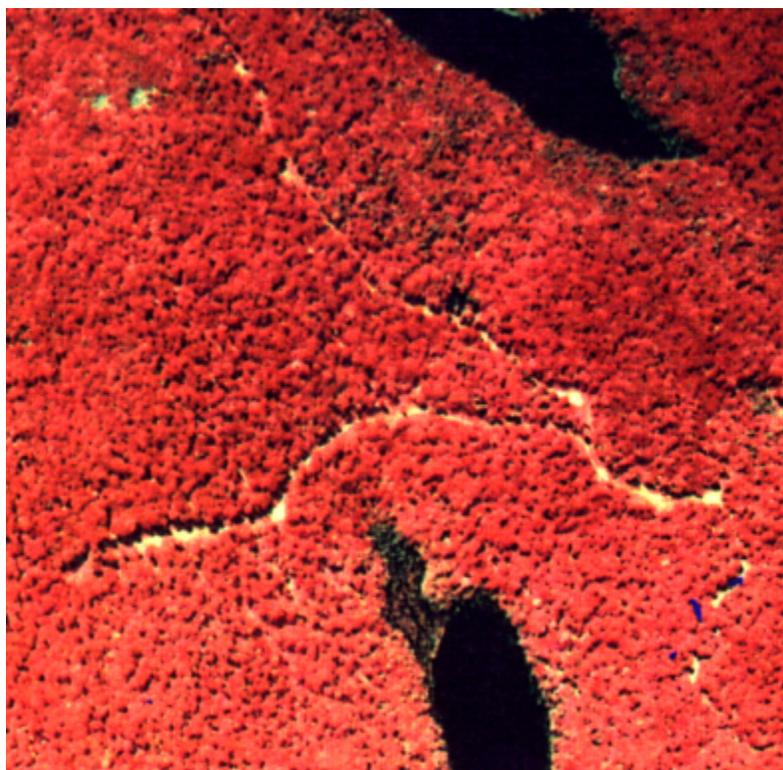


Figure 1. Photo aérienne infrarouge d'un secteur de coupe venant d'être traité par jardinage par pied d'arbre. On peut facilement y observer la récurrence systématique des ouvertures.

Bien qu'un modèle de simulation par tiges dépendant de la distance permettrait de distinguer ces deux modes distincts de répartition des ouvertures après jardinage, l'effort nécessaire pour la caractérisation spatiale des tiges lors des inventaires proscrit l'utilisation de ce type de modèle à des fins opérationnelles.

Objectifs de recherche

Le premier objectif de recherche de ce projet vise donc à développer un algorithme qui serait inclut dans un module de *Cohorte* afin que celui-ci puisse ajuster la compétition subie par une tige après une CJ+ d'un prélèvement d'un niveau donné, tout en tenant compte de l'organisation sociale des tiges du peuplement évalué. Plus spécifiquement, l'objectif de ce travail de recherche est de quantifier la différence dans le changement dans l'indice de compétition avant et après traitement entre un prélèvement spatialement systématique (CJ) et un prélèvement avec dégagement accru des tiges élites (CJ+), et selon la quantité de tiges élites à dégager, le niveau de prélèvement, le type de prélèvement (par le haut, équilibré, par le bas) et la surface terrière initiale, à partir des indices de compétition spatiaux et aspatiaux.

Approche analytique

Pour répondre aux objectifs de recherche, nous utilisons une approche analytique par simulation. En utilisant des grandes placettes-échantillons dans lesquelles tous les arbres sont cartographiés, il est possible de simuler les traitements de CJ et de CJ+ en faisant varier les paramètres de prélèvement (intensité, type, avec dégagement accru).

Hypothèses

Les hypothèses de recherche sont :

- a) Toute autre chose étant égale, la différence de réduction dans l'indice de compétition spatial pour les tiges élites et les tiges de bourrage est indépendante du DHP de la tige. De même, on s'attend que cette indépendance au DHP soit maintenue, quelque soit le prélèvement, la surface terrière en tiges élites, la surface terrière initiale et la différence du diamètre moyen après coupe;

- b) Pour tous niveaux de prélèvement et surfaces terrières initiales confondus, les indices de compétition spatiaux des tiges élites présenteront une réduction plus importante après un prélèvement avec dégagement accru des tiges élites (CJ+) qu'après un prélèvement balancé (CJ).
- c) L'inverse sera observé pour les tiges de bourrage.
- d) Toute autre chose étant égale, la différence de changement dans l'indice de compétition spatial sera moins importante à des forts niveaux de prélèvement. Ainsi, plus le prélèvement sera important, moins la réduction de compétition sera différente entre un prélèvement avec dégagement accru des tiges élites (CJ+) et un prélèvement balancé (CJ) ;
- e) Toute autre chose étant égale, la différence de réduction dans l'indice de compétition spatial sera moins importante pour les faibles surfaces terrières initiales;
- f) Toute autre chose étant égale, la différence de réduction dans l'indice de compétition spatial pour les tiges élites sera amoindrie plus la surface terrière en tiges élites à dégagée sera importante;
- g) Toute autre chose étant égale, la différence de réduction dans l'indice de compétition spatial pour les tiges élites sera plus importante pour les prélèvements par le haut que les prélèvements équilibrés dans les classes de tailles.

2-Méthodologie

Jeu de données

Quarante-neuf grandes placettes-échantillons (PÉ) ont été utilisées dont 36 possèdent un rayon de 28,21 m (0.25 ha) et 13 avec un rayon de 30 m (0.28 ha). Ces PÉs sont localisées dans l'Outaouais (Figure 2) et proviennent de différentes forêts dominées par des feuillus tolérants (Tableau 1). Le diamètre au DHP (1.33 m) a été pris pour tous les arbres ayant 9.1 cm et plus. De plus, tous les arbres à l'intérieur de ces PÉs ont été localisés et géoréférencés. Pour ce faire, nous avons utilisé un mesureur à l'ultrason (Haglof, VERTEX III avec un transpondeur T3 $\pm 1^\circ$) pour mesurer la distance de l'arbre au centre de la PÉ et une boussole SUUNTO à visée intérieure (Modèle KB14, $\pm 0.5^\circ$) pour l'angle à partir du point centre de la PÉ. Dans le cas des 13 PÉs de rayon de 30 m, des quatre points intermédiaires de relais ont été positionnés en « X » par rapport au centre de la PÉ à 20 m de celui-ci. Dans le cas d'une pente supérieure à 30% entre l'arbre et le centre de la PÉ (erreur < 4.4 cm sur 1 m), une correction pour la pente était apportée pour obtenir la distance horizontale.



Figure 2. Les placettes-échantillons ayant servi pour ce projet sont dans l'Outaouais, dans la partie ouest du domaine bioclimacique de l'érablière à bouleau jaune.

Afin d'éviter un effet de bordure lors du calcul de l'indice de compétition des tiges à l'intérieur d'une PÉ, il a été nécessaire de générer une zone tampon à l'extérieur de la PÉ. Pour ce faire, quatre carrés étaient sous-échantillonnés à l'intérieur de la PÉ (Figure 3; carrés 1, 2, 3 et 4) et chacun était ensuite réparti 3 fois aléatoirement autour de la PÉ (Figure 3; carrés avec le chiffre « primé ») de façon à obtenir une représentation égale entre les 4 carrés. Les portions de cercle à l'intérieur de la PÉ (Figure 3; a, b, c, d, e, f, g, et h) étaient par la suite « collés » par-dessus les carrés de la zone tampon pour reconstituer la PÉ intégrale. Cette étape était reproduite pour chaque simulation, générant une nouvelle zone tampon autour de chacune des PÉs.

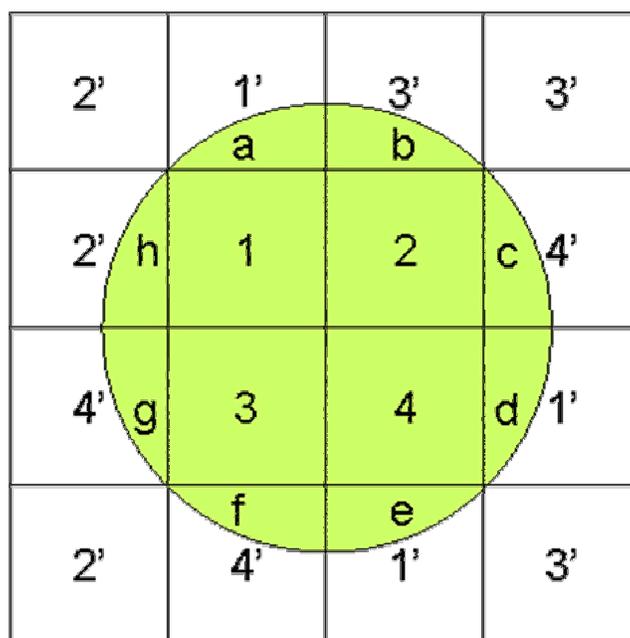


Figure 3. Création d'une zone tampon autour de la parcelle-échantillon pour éviter l'effet de bordure lors du calcul des indices de compétition. Les zones avec un nombre primé reproduisent la zone avec le même nombre à l'intérieur de la PÉ.

Calcul de l'indice de compétition

Un indice spatial de compétition a été calculé pour chaque tige avant et après avoir appliqué le prélèvement d'un traitement. Cet indice de compétition évalue la surface terrière des tiges ayant un dhp plus élevé que le $\frac{3}{4}$ du dhp de la tige sujet dans un rayon de 10 m autour de celle-ci (Équation 1).

$$\text{Équation 1 : } IC = \sum_i St_{i(dhpi > 0.75dhps)}$$

où $St_{i(dhpi > 0.75dhps)}$: surface terrière de la tige compétitrice « i » dans un rayon de 10 m autour de la tige sujet « s » et de DHP plus grand que 0.75 le DHP de la tige sujet. L'unité de cet indice est en m^2/ha .

Modélisation du prélèvement balancé

Pour ce projet, un prélèvement balancé est un prélèvement aléatoire sur toute la PÉ (et la zone tampon), mais réparti également du point de vue spatial. Pour ce faire, dans un premier temps, la PÉ est subdivisée en tuiles de taille visant à permettre de récolter une tige par tuile en moyenne. Conséquemment, le nombre de tuiles par PÉ est donc régi par la densité en tiges de la PÉ et par le niveau de prélèvement. Le nombre de tuiles est le carré le plus près de la surface terrière à récolter (surface terrière de la PÉ * niveau de prélèvement) divisé par la surface terrière moyenne des tiges faisant parties de la PÉ.

Suite à cette séparation du territoire, des tuiles sont aléatoirement examinées par le programme. Lors de la visite d'une tuile par le programme, une ou des tiges dans celle-ci sont sélectionnées toujours aléatoirement, de façon à simuler un prélèvement proche du niveau global de prélèvement. Pour ce faire, un niveau local de prélèvement est établi (à l'échelle de la tuile). Un autre arbre est sélectionné pour récolte tant et aussi longtemps que le niveau local de prélèvement n'est pas dépassé ou qu'il n'a plus d'arbres dans la tuile. Cependant, le prélèvement à l'échelle de la tuile ne peut dépasser de $0.5 m^2/ha$ du prélèvement attendu dans la tuile. Lorsque le niveau de prélèvement local est dépassé, on compare si la différence absolue est plus élevée avant la récolte de ce dernier arbre qu'après sa récolte. Si la différence est moins élevée après, ce dernier arbre est maintenu parmi le groupe des récoltés, sinon, il est exclu du groupe des arbres récoltés dans la tuile. Le programme passe ensuite aléatoirement à une autre tuile. Le programme garde cependant en mémoire la somme des prélèvements effectués à mesure qu'il progresse dans la PÉ. Si le prélèvement total est plus important qu'attendu, alors le programme prélève moins (ou pas du tout) dans la tuile qu'il est en train d'évaluer. Par contre, si le prélèvement total est moins important qu'attendu, le programme augmentera légèrement son prélèvement dans la tuile. Le programme progresse ainsi en visitant toutes les tuiles, jusqu'à ce qu'il atteigne le prélèvement prescrit, à $\pm 0.5 m^2/ha$.

En agissant de la sorte, il est arrivé pour certaines simulations que le prélèvement total respecte la condition $\pm 0.5 \text{ m}^2/\text{ha}$ du prélèvement prescrit pour l'ensemble PÉ et bordure mais que cela ne soit pas le cas pour la zone limitée à l'intérieur de la PÉ (zones 1, 2, 3, 4, a, b, c, d, e, f, g, et h de la Figure 3). Pour éviter cela, le prélèvement est alors réévalué pour l'intérieur de la PÉ seulement (excluant la zone tampon). Trois résultats peuvent se produire à ce moment :

- 1- Le taux de récolte n'est pas atteint : Nous sélectionnons alors des tiges aléatoirement à l'intérieur de la PÉ une par une jusqu'à ce que nous atteignons le taux de prélèvement. Lorsque la dernière tige récoltée fait dépasser le taux de prélèvement, on compare alors si on était plus près du taux de prélèvement avant récolte de cette tige qu'après pour savoir si elle sera incluse ou exclue des tiges récoltées.
- 2- Le taux de récolte est dépassé de plus de $0.5 \text{ m}^2/\text{ha}$: Les tuiles sont triées en ordre décroissant selon la surface terrière récoltée et on retire de l'ensemble des tiges récoltées la plus grosse tige de la tuile qui possède le plus fort taux de prélèvement. Si après avoir enlevé cette tige le taux de prélèvement est toujours dépassé de $0.5 \text{ m}^2/\text{ha}$, la deuxième tuile ayant le plus fort taux de prélèvement est sélectionnée et on retire l'arbre récolté le plus gros de cette tuile et ainsi de suite.
- 3- Le taux de prélèvement est dépassé mais de moins de $0.5 \text{ m}^2/\text{ha}$: Nous quittons alors le module de récolte.

Un exemple de la répartition spatiale des arbres récoltés à l'aide de cet algorithme de récolte pour une PÉ est présenté en Figure 4 b).

Modélisation du prélèvement avec dégagement accru des tiges élites

Pour ce projet, un prélèvement avec dégagement accru des tiges élites est un prélèvement qui s'intensifie autour des tiges élites de la PÉ (et la zone tampon). Ainsi, la répartition spatiale du prélèvement qui en résulte a tendance à être plus agrégée. Pour simuler ce type de dégagement, comme il est désiré d'avoir une répartition bien distribuée de tiges élites dans le peuplement, nous avons dans un premier temps divisé la PÉ (plus la zone tampon) en tuiles. Le nombre de tuiles est égal au nombre de tiges élites à être sélectionnées dans la PE (plus la zone tampon). Par la suite, on évalue la valeur de la compétition produite par chaque tige de bourrage dans un rayon de 10 m autour de chaque tige élite à l'aide de l'Équation 2 :

$$\text{Équation 2 : } (dhp_{\text{comp}} / (\text{distance}_{\text{comp-sujet}} * dhp_{\text{sujet}}))$$

où dhp_{sujet} : dhp de la tige élite
 dhp_{comp} : dhp de la tige compétition
 $\text{distance}_{\text{comp-sujet}}$: distance entre la tige élite et compétition

Par la suite, le programme sélectionne aléatoirement une tige élite et prélève la tige produisant la plus forte compétition pour cette tige élite. Si le niveau de prélèvement n'est pas atteint localement (dans la tuile) suite à cette première sélection de tige du bourrage à prélever, une autre tige de bourrage, soit la suivante ayant l'IC le plus élevé, est sélectionnée. Cependant, afin de maintenir une structure plus équilibrée, cette seconde tige sera préférentiellement dans une autre classe de diamètre (les classes de dhp étant : 9.1-19.0, 19.1-29.0, 29.1-39.0, 39.1+) que la tige de bourrage qui avait été sélectionnée précédemment. Cette séquence est répétée tant et aussi longtemps que le niveau de prélèvement n'est pas atteint ou qu'il ne reste plus aucune tiges de bourrage autour de la tige élite. On procède alors à la sélection aléatoire de la tige élite suivante, jusqu'à ce que le niveau de prélèvement global soit atteint. Si à la fin de ce processus le niveau de prélèvement n'est toujours pas atteint, i.e., plus aucune tige de bourrage reste dans le rayon de 10 m, une sélection aléatoire au sein des tiges restantes du bourrage est effectuée pour compléter le prélèvement (ce cas arrive en fait lorsque la surface terrière en tiges élites est faible et que le niveau de prélèvement est élevé).

Un exemple de la répartition spatiale des arbres récoltés à l'aide de cet algorithme de récolte pour une PÉ est présenté en Figure 4 c).

Le code Visual Basic de l'algorithme de récolte partielle pour le prélèvement balancé et avec dégagement accru des tiges élites utilisé dans cette étude est présenté en Annexe 1.

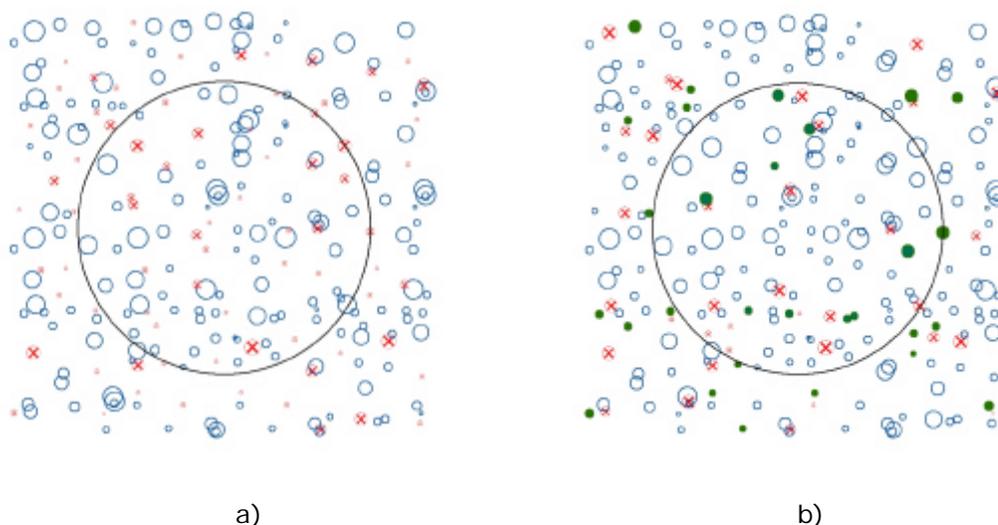


Figure 4. Répartition spatiale des arbres avant et après traitement dans une des PÉs traitées par a) CJ et par b) CJ+ par l'algorithme de récolte partielle utilisé dans cette étude. Les arbres sont représentés par des cercles. Les cercles avec le contour en bleu sont les arbres résiduels après traitement, les arbres en rouge avec un « X » à l'intérieur sont les arbres prélevés et les arbres en vert plein sont les tiges élites. Le prélèvement est de 30%. La surface terrière en tiges élites dans la CJ+ est de 1m²/ha.

Simulations

Le prélèvement balancé et celui avec dégagement accru sur les tiges élites a été simulé sur les 49 PÉs avec 5 niveaux de prélèvement (20%, 25%, 30%, 35% et 40%) de la surface terrière et 8 niveaux de surface terrière en tiges élites (0.5, 1, 1.5, 2, 3, 5, 7 et 9 m²/ha). Chaque combinaison de prélèvement/surface terrière en tiges élites/PÉ est simulée 100 fois (soit 5 * 8 * 49 * 100 pour la CJ+ et 5 * 49 * 100 pour la CJ, donnant 220 500 simulations). Pour chacun des 100 répliqués de traitement (c.-à-d. 1 CJ et 8 CJ+), une nouvelle zone tampon était créée. Une fois les simulations de prélèvement effectuées, l'indice de compétition avant et après traitement est calculé pour tous les arbres de la PÉ (donc excluant la zone tampon) n'ayant pas été récoltés, en maintenant la distinction entre les tiges élites et les tiges de bourrage en ce qui concerne le prélèvement avec dégagement accru des tiges élites.

Analyses

Les hypothèses ont été testées à l'aide de l'analyse de variance et de la régression linéaire. Ce test est fait séparément pour les tiges élites et les tiges de bourrage.

Le modèle statistique testé est le suivant :

$$IC_{\text{avant}} - IC_{\text{après}} = \text{DHP} + \text{Prélèvement} + \text{St initiale} + \text{St élites} + \text{DHP moyen}$$

- $IC_{\text{avant}} - IC_{\text{après}}$: Différence entre la CJ et la CJ+ de la différence d'IC avant et après traitement pour une tige. Une valeur supérieure à zéro indiquera que la CJ+ a réduit la compétition plus fortement que la CJ et vice et versa. On s'attend donc que cet indice soit positif pour les tiges élites et négatif pour les tiges de bourrage;
- DHP : DHP de la tige évaluée
- Prélèvement : % de la surface terrière prélevée (20, 25, 30, 35, 40%)
- St initiale : Surface terrière avant traitement;
- St élites : Surface terrière en tiges élites à dégager;
- DHP moyen) : Différence entre la moyenne quadratique du DHP des tiges de la PÉ avant et après CJ+. Cette variable exprime le type de prélèvement; une valeur positive caractérisera un prélèvement par le haut et vice et versa.

Dans cette analyse, chaque arbre/placette/traitement est considéré comme un élément statistique. Ainsi, avant de procéder à l'analyse, pour un arbre d'une PÉ et pour chaque traitement, la moyenne des valeurs des simulations lorsqu'il est soit élite ou soit bourrage est utilisée. À titre d'exemple, pour l'arbre « A » de la PÉ « P » dans le traitement CJ+ avec 25% de prélèvement et 1.5 m²/ha en surface terrière en tiges élites, il existe 100 simulations. De ces 100 simulations, il y a un nombre « X » dans lesquelles l'arbre « A » est une tige élite, un nombre « Y » dans lesquelles l'arbre « A » est une tige de bourrage et nombre « Z » dans lesquelles l'arbre « A » fait partie des tiges récoltées. Ainsi, pour nos analyses, nous utiliserons la moyenne des $IC_{\text{avant}} - IC_{\text{après}}$ de cet arbre lorsqu'il est soit élite (moyenne des « X » simulations), soit bourrage (moyenne des « Y » simulations).

Le test est un modèle complet, testant aussi pour tous les effets d'interaction. Les analyses ont été réalisées à l'aide du progiciel R (R Development Core Team 2009). Le modèle final de prédiction de la différence de compétition retenu a été sélectionné en utilisant le coefficient de l'information de Akaike (AIC). Nous avons testé avec la régression multiple le modèle qui serait susceptible de mieux prédire l'indice de différence de changement de compétition après traitement tout en demeurant le plus parcimonieux. Pour ce faire, nous

avons utilisé la sélection pas-à-pas des variables et nous avons par la suite retenu les termes du modèle qui contribuaient significativement à l'augmentation de la prédiction.

3 – Résultats

Description du jeu de données

Les PÉs sont composés en majorité de feuillus tolérants (79%) et de feuillus peu tolérants (12%) ces derniers pouvant parfois dominer dans certaines PÉs (Tableau 1). La surface terrière observée varie de 16 à 35 m²/ha avec une moyenne de 23.7 m²/ha avec des moyennes quadratiques du DHP variant de 17 à 34 cm, la moyenne étant de 25.12 cm. Nous avons donc une bonne diversité de structure de peuplements pour notre évaluation.

Tableau 1. Composition et structure des placettes-échantillons ayant servi pour ce projet.

No. PÉ	Superficie (ha)	Surface terrière (m ² /ha)	DHP ^a moyen (cm)	Feuillus ^b durs	Feuillus ^c peu tolérants	Feuillus ^d into-tolérants	Autres ^e feuillus	SEPM ^f	Pruche	Autres ^g résineux
0_1	0.25	27.33	34.29	99%	1%	0%	0%	0%	0%	0%
0_2	0.25	27.58	29.78	100%	0%	0%	0%	0%	0%	0%
0_3	0.25	25.18	29.20	97%	3%	0%	0%	0%	0%	0%
1_1	0.25	26.75	33.47	100%	0%	0%	0%	0%	0%	0%
1_2	0.25	31.84	31.22	65%	0%	9%	1%	0%	31%	0%
1_3	0.25	22.40	27.84	85%	4%	0%	4%	0%	7%	0%
10_1	0.25	21.13	25.94	92%	7%	1%	0%	0%	0%	0%
10_2	0.25	22.76	22.43	95%	1%	10%	0%	1%	2%	0%
10_3	0.25	21.31	22.58	82%	0%	15%	1%	5%	6%	3%
2_1	0.25	34.50	31.04	84%	11%	2%	1%	0%	4%	0%
2_2	0.25	20.25	26.76	99%	0%	0%	1%	0%	0%	0%
2_3	0.25	31.58	32.36	71%	10%	0%	0%	0%	19%	0%
3_1	0.25	16.36	28.52	94%	6%	0%	0%	0%	0%	0%
3_2	0.25	16.11	24.28	93%	7%	0%	0%	0%	0%	0%
3_3	0.25	20.10	20.32	83%	4%	5%	7%	0%	0%	0%
4_1	0.25	18.71	18.77	74%	23%	3%	1%	0%	0%	0%
4_2	0.25	22.81	29.22	99%	0%	1%	1%	0%	0%	0%
4_3	0.25	22.47	26.88	100%	0%	0%	0%	0%	0%	0%
5_1	0.25	25.95	27.78	100%	0%	0%	0%	0%	0%	0%
5_2	0.25	24.09	33.83	100%	0%	0%	0%	0%	0%	0%
5_4	0.25	24.79	25.03	85%	0%	0%	1%	0%	14%	0%
6_1	0.25	23.40	30.14	98%	0%	10%	0%	0%	1%	0%

6_2	0.25	18.55	22.28	90%	6%	25%	0%	2%	0%	0%
6_3	0.25	24.34	20.86	72%	2%	9%	0%	18%	0%	3%
7_1	0.25	20.59	19.81	35%	54%	9%	1%	2%	0%	0%
7_2	0.25	18.03	17.34	77%	12%	6%	4%	1%	0%	0%
7_3	0.25	20.57	21.86	95%	1%	20%	0%	0%	0%	0%
8_1	0.25	21.09	22.90	94%	0%	0%	0%	6%	0%	0%
8_2	0.25	19.69	19.15	50%	0%	52%	0%	0%	2%	0%
8_3	0.25	24.65	20.82	64%	0%	31%	0%	0%	0%	0%
9_1	0.25	27.42	24.29	88%	12%	8%	0%	0%	0%	0%
9_2	0.25	31.57	27.70	70%	26%	0%	4%	0%	0%	0%
9_3	0.25	24.77	28.08	88%	11%	0%	2%	0%	0%	0%
F_0	0.25	24.32	23.27	31%	52%	6%	3%	4%	0%	3%
F_1	0.25	20.33	21.12	91%	2%	8%	2%	5%	0%	0%
F_5	0.25	23.64	23.79	55%	31%	11%	4%	4%	0%	2%
Mrc1	0.28	29.32	26.98	79%	21%	0%	1%	0%	0%	0%
Mrc10	0.28	23.74	24.45	95%	0%	2%	0%	0%	3%	0%
Mrc11	0.28	23.68	24.00	82%	12%	0%	7%	0%	0%	0%
Mrc12	0.28	21.42	21.89	19%	71%	5%	1%	1%	0%	3%
Mrc13	0.28	24.40	26.20	98%	1%	0%	1%	0%	0%	0%
Mrc2	0.28	19.84	18.31	35%	48%	0%	17%	0%	0%	0%
Mrc3	0.28	22.70	25.17	73%	4%	19%	1%	0%	8%	0%
Mrc4	0.28	20.88	22.46	32%	56%	1%	10%	0%	0%	0%
Mrc5	0.28	22.38	22.51	57%	18%	16%	4%	3%	1%	0%
Mrc6	0.28	29.88	24.93	85%	3%	0%	1%	2%	10%	0%
Mrc7	0.28	28.67	26.23	84%	4%	12%	1%	0%	0%	0%
Mrc8	0.28	24.08	21.71	99%	0%	3%	0%	0%	0%	0%
Mrc9	0.28	23.59	21.14	28%	67%	3%	3%	0%	0%	0%
Moyenne		23.70	25.12	79%	12%	6%	2%	1%	2%	0%

^a Moyenne quadratique;

^b Érable à sucre, érable rouge, bouleau jaune, hêtre à grandes feuilles;

^c Cerisier tardif, chêne rouge, frêne d'Amérique, tilleul d'Amérique;

^d Bouleau à papier, bouleau gris, peuplier à grandes dents, peuplier baumier, peuplier faux-tremble, cerisier de Pennsylvanie;

^e Frêne noir, ostryer de Virginie, noyer cendré, orme d'Amérique, orme rouge;

^f Sapin baumier, épinette blanche, épinette noire, épinette rouge;

^g Pins blancs, pins rouges, thuya.

Bien que nous aurions voulu avoir un jeu de données ne présentant pas de lien entre la moyenne quadratique du DHP et la surface terrière, on observe une relation positive entre

ces deux variables (Figure 5). En général, on observe néanmoins une bonne fourchette de DHP moyen pour chaque classe de surface terrière de 4 m²/ha.

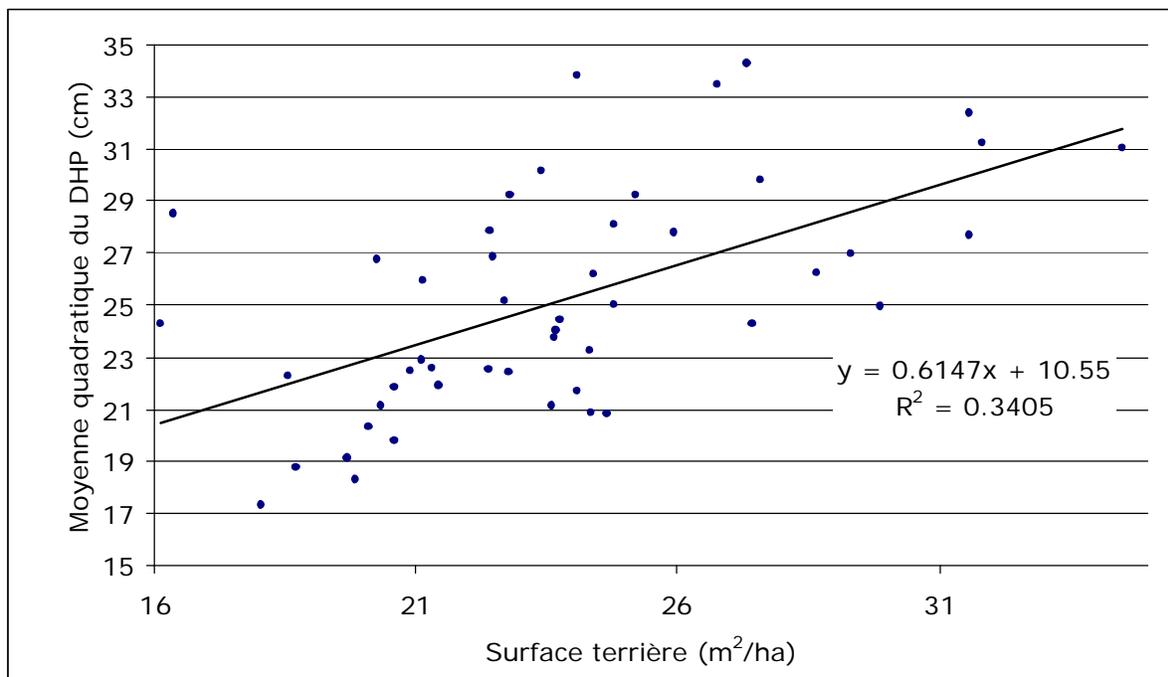


Figure 5. Relation entre la moyenne quadratique du DHP et la surface terrière des PÉs.

Moyenne générale des différences de changement de compétition après traitement

Nous avons voulu dans un premier temps tester si la moyenne générale de notre indice de différence de changement de compétition après traitement était différente de zéro en ce qui concerne les tiges élites et les tiges de bourrage. En ce qui concerne le bourrage, nous trouvons que la moyenne générale de notre indice est significativement différente de 0 ($t = 19.024$, $df = 248490$, $P < 2.2e-16$) avec une valeur positive de 0.0133556 m²/ha. Ce résultat étonne car nous nous attendions à ce que le bourrage obtienne une valeur négative. Pour ce qui est des tiges élites, la moyenne générale de notre indice est significativement différente de 0 ($t = 322.786$, $df = 223148$, $P < 2.2e-16$) avec une valeur positive de 0.3848771 m²/ha.

Type de prélèvement

Le fait que l'indice de différence de changement de compétition après traitement soit significativement différente de zéro et positive indiquait que quelque chose d'autre que juste le niveau de prélèvement et la surface terrière en tiges élites pouvait agir sur la compétition. Nous avons donc voulu inclure l'effet du changement dans l'organisation sociale des tiges après traitements entre la CJ et la CJ+. Pour ce faire, nous avons testé si la moyenne quadratique des tiges des PÉs avant et après traitement était différente pour la CJ et pour la CJ+. Effectivement, nos résultats montrent qu'après la CJ, la moyenne n'est pas significativement différente qu'avant (-0.10 cm +/- 0.33 cm) alors que dans la CJ+, la moyenne quadratique est significativement plus petite après traitement qu'avant (-2.12 cm +/- 1.32 cm). Cette différence pour la CJ+ a tendance à s'accroître avec le taux de prélèvement (Figure 6). Nous avons donc ajouté une variable qui tient compte de la différence de changement dans la structure diamétrale entre les deux traitements (CJ et CJ+). Avec cet indicateur, plus la différence de moyenne de DHP après traitement est élevée, plus la CJ+ a un prélèvement axé sur les grosses tiges comparativement à la CJ.

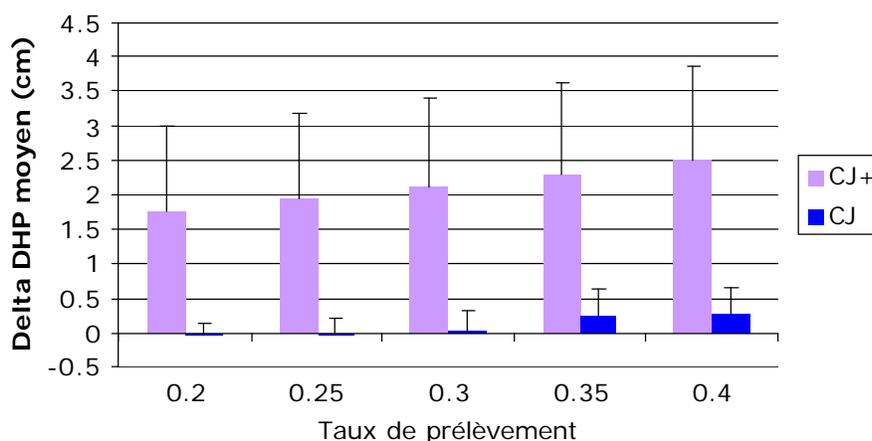


Figure 6. Différence de la moyenne quadratique avant et après traitement pour la CJ et la CJ+.

Effets simples sur l'indice de différence de changement de compétition après traitement

Nous avons testé les effets simples des variables *DHP de la tige*, *surface terrière initiale*, *prélèvement*, *surface terrière en tiges élites*, et *différence en DHP moyen entre avant et après traitement* sur l'indice de changement de différence de la compétition après traitement

à l'aide de la régression linéaire. Même si on constate que toutes les variables ont un effet sur l'indice, tant pour les tiges élites que les tiges de bourrage, plusieurs de ces effets significatifs ne sont dus qu'au grand nombre de réplicats ($n > 200000$) (Tableau 2). En effet, parmi tous les effets simples, seule la surface terrière en tiges élites semble avoir vraiment un pouvoir explicatif sur l'indice de changement de différence de la compétition après traitement, et cela, pour les tiges élites seulement. Pour les tiges de bourrage, cette dernière variable demeure la plus significative parmi toutes.

Tableau 2. Tableau de l'analyse de variance de la régression linéaire entre les variables du modèle et la différence de changement de compétition après traitement.

Type de tiges	Variable	Ordonnée	Coefficients			Modèle			
			<i>P</i>	Pente	<i>P</i>	<i>F</i>	<i>DL</i>	<i>P</i>	<i>R</i> ²
Élite	DHP de la tige	0.364	<0.001	0.001	<0.001	66.53	223147	<0.001	0.000
	St initiale	-0.144	"	0.022	"	5483	"	"	0.024
	Prélèvement	0.133	"	0.841	"	2513	"	"	0.011
	(St en tiges élites) ^{0.769}	-0.036	"	0.729	"	98610	"	"	0.307
	(Delta DHP moyen) ²	0.372	"	0.003	"	99.37	"	"	0.000
Bourrage	DHP de la tige	0.052	<0.001	-0.002	<0.001	744.9	248489	<0.001	0.003
	St initiale	0.181	"	-0.007	"	1569	"	"	0.006
	Prélèvement	0.093	"	-0.266	"	719.4	"	"	0.003
	(St en tiges élites) ^{0.769}	0.093	"	-0.123	"	7155	"	"	0.028
	(Delta DHP moyen) ²	0.002	"	0.003	"	230.7	"	"	0.001

Pour l'effet du DHP de la tige, on observe qu'il n'y a pas de relation pour les tiges élites, si ce n'est une baisse de l'indice pour les très grosses tiges (Figure 7). Pour les tiges de bourrage, il n'y a pas de tendance observable (Figure 8).

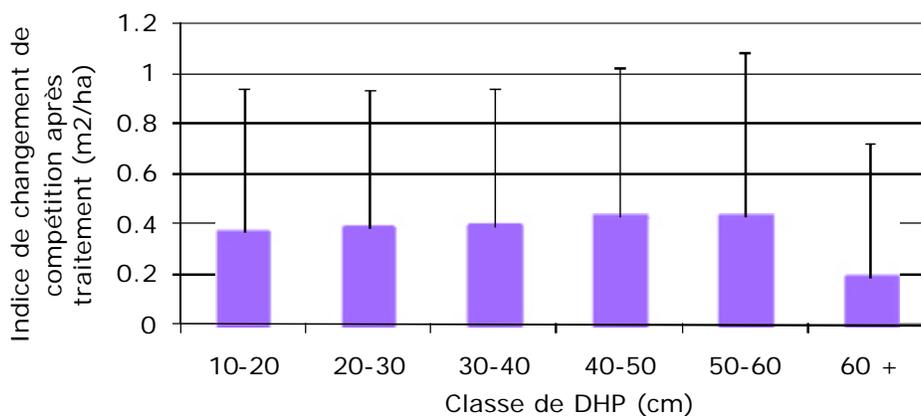


Figure 7. Relation entre le DHP de la tige et l'indice de la différence de changement de compétition après traitement pour les tiges élites.

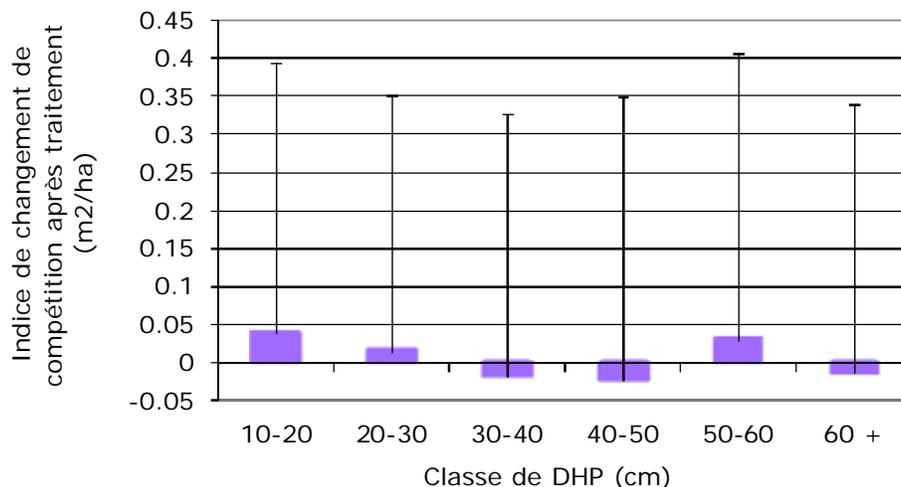


Figure 8. Relation entre le DHP de la tige et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.

L'indice de changement de la différence en compétition après traitement augmente légèrement en fonction de la surface terrière initiale pour les tiges élites (Figure 9) alors qu'il diminue pour les tiges de bourrage (Figure 10). Pour les deux types de tiges, il s'agit de la deuxième variable, après la surface terrière en tiges élites, qui explique le plus les variations de l'indice de changement de la différence en compétition après traitement.

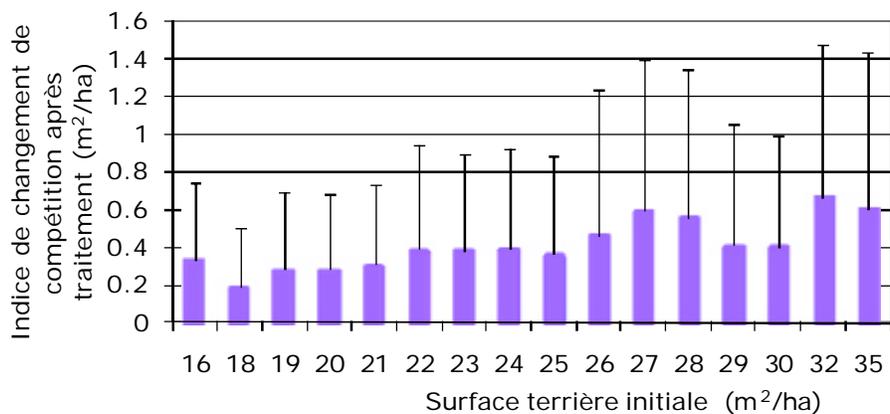


Figure 9. Relation entre la surface terrière initiale et l'indice de la différence de changement de compétition après traitement pour les tiges élites.

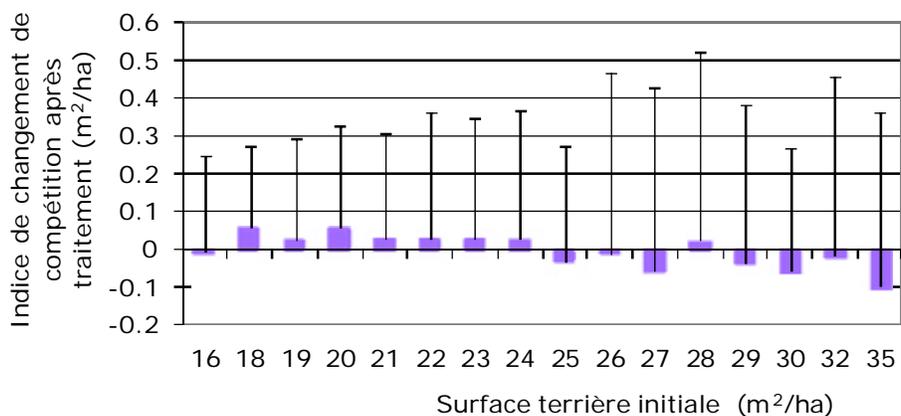


Figure 10. Relation entre la surface terrière initiale et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.

L'indice de changement de la différence en compétition après traitement augmente légèrement en fonction du taux de prélèvement pour les tiges élites (Figure 11) alors qu'il diminue pour les tiges de bourrage (Figure 12).

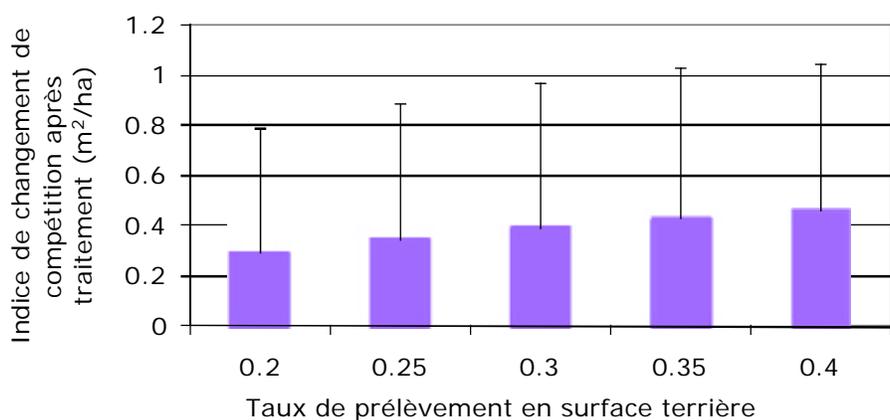


Figure 11. Relation entre le taux de prélèvement en surface terrière et l'indice de la différence de changement de compétition après traitement pour les tiges élites.

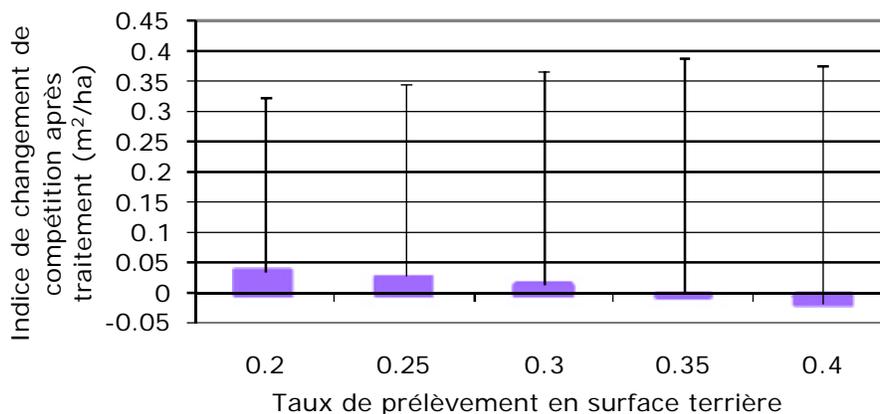


Figure 12. Relation entre le taux de prélèvement en surface terrière et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.

La surface terrière en tiges élites est assurément la variable la plus importante pour modéliser le dégagement accru des tiges élites tel qu'exprimé par l'indice de changement de la différence en compétition après traitement. Cet effet décroît de façon non-linéaire pour se stabiliser après $3 \text{ m}^2/\text{ha}$ (Figure 13). Il est surtout apparent pour des surfaces terrières en tiges élites inférieures à $3 \text{ m}^2/\text{ha}$. Afin de rendre linéaire cette relation pour nos analyses de régression, nous avons transformé la variable à l'aide d'une fonction de puissance ($Y=St$ en tiges élites)^{0.789}).

Cette relation non-linéaire entre la surface terrière en tiges élites et l'indice de la différence de changement de compétition après traitement est à l'inverse, pour les tiges de bourrage, de celle détectée pour les tiges élites (Figure 14). Elle se stabilise plus tôt soit autour d'une asymptote à partir de $2 \text{ m}^2/\text{ha}$. Ainsi, on observe que les tiges de bourrage seront plus compétitionnées en CJ+ qu'en CJ lorsque la surface terrière en tiges élites sera inférieure à $1.5 \text{ m}^2/\text{ha}$, après quoi, les tiges de bourrage deviendront moins compétitionnées en CJ+ qu'en CJ.

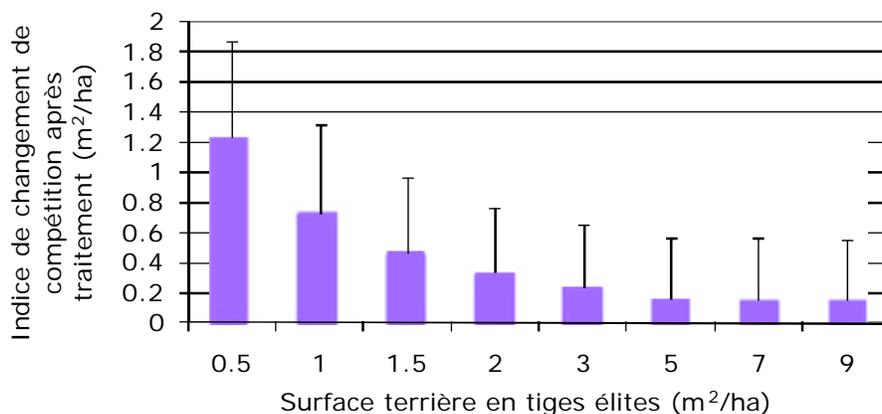


Figure 13. Relation entre la surface terrière en tiges élites et l'indice de la différence de changement de compétition après traitement pour les tiges élites.

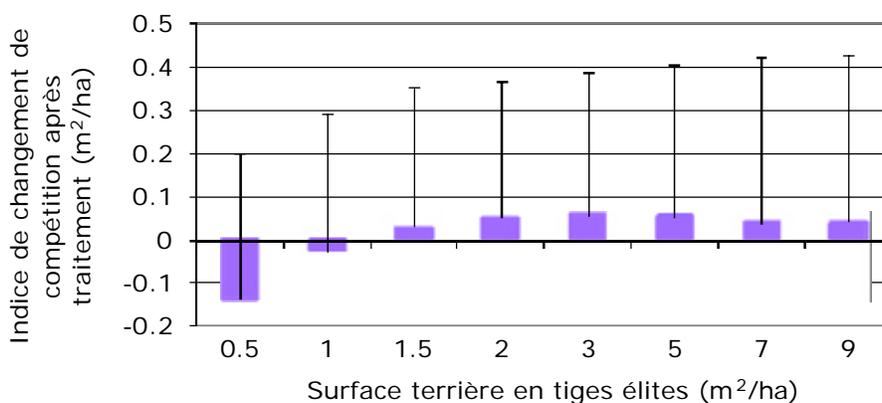


Figure 14. Relation entre la surface terrière en tiges élites et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.

Pour les tiges élites, on observe que l'indice de la différence de changement de compétition après traitement est plus élevée lorsqu'il n'y a pas de différence dans le type de prélèvement (0 cm), puis devient à son plus bas lorsque cette différence est de 1 cm pour augmenter ensuite (Figure 15). Pour les tiges de bourrage, l'indice de la différence de changement de compétition après traitement est négatif lorsqu'il n'y a pas de moyenne de DHP après traitement et il devient positif aussitôt que la différence est de 1 cm (Figure 16).

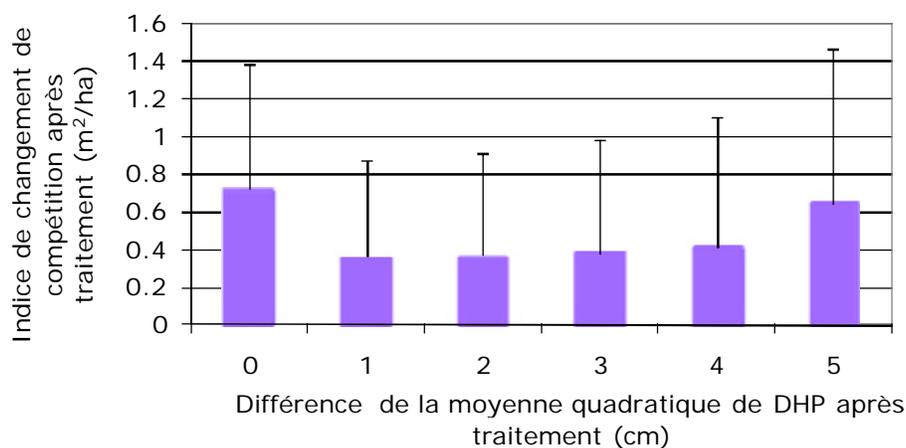


Figure 15. Relation entre la différence de la moyenne quadratique de DHP après traitement et l'indice de la différence de changement de compétition après traitement pour les tiges élites.

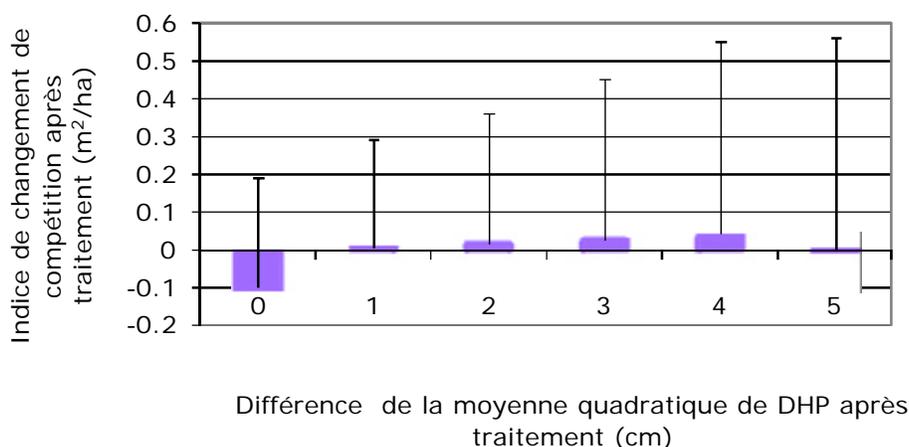


Figure 16. Relation entre la différence de la moyenne quadratique de DHP après traitement et l'indice de la différence de changement de compétition après traitement pour les tiges de bourrage.

Modèle prédictif de l'indice de différence de changement de compétition après traitement

Pour le modèle pour prédire l'indice de changement de différence dans la compétition après traitement pour les tiges élites, la sélection pas-à-pas du modèle de régression multiple montre que les effets simples les plus importants (St en tiges élites, St initiale et prélèvement, ainsi que leurs interactions ont été sélectionnés en premier avant d'inclure les

autres variables (DHP de la tige, Différence de DHP moyen après traitement et sa valeur au carré) (Tableau 3).

Tableau 3. Étape de la sélection pas-à-pas des termes du modèle de régression multiple pour les tiges élites

Pas	R ²	Amélioration du R ²	Termes
1	0.3065	0.3065	St élites
2	0.3342	0.0277	St élites + St initial
3	0.3598	0.0257	St élites + St initial + St élites * St initial
4*	0.3708	0.011	St élites + St initial + St élites * St initial + Prélèvement
5	0.3739	0.0031	St élites + St initial + St élites * St initial + Prélèvement + St élites * Prélèvement
6	0.3741	0.0002	St élites + St initial + St élites * St initial + Prélèvement + St élites * Prélèvement + St initial * Prélèvement

* Notre modèle final pour les tiges élites est donc :

$$\text{Équation 3. ICDCAT}^* = \text{St Élites} + \text{St initial} + \text{St Élites} * \text{St initial} + \text{Prélèvement}$$

*Indice de changement de différence dans la compétition après traitement

Ce modèle final explique une portion importante de la variabilité ($R^2=0.3708$) mais il demeure toutefois une bonne proportion de la variance inexpliquée, que les autres variables ou leurs interactions ne permettent pas de prédire. La majeure partie de la variance expliquée par notre modèle demeure celle expliquée par la surface terrière en tiges élites (Tableau 3). Il est intéressant de noter que l'inclusion d'un effet d'interaction entre la surface terrière en tiges élites et la surface terrière initiale a amené à changer le signe de la relation pour les deux effets principaux de cette interaction dans le modèle final (Tableau 4). L'analyse de variance montre les sources d'erreurs du modèle et celles associées à chaque terme (Tableau 5).

Tableau 4. Coefficients des termes utilisés dans le modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges élites

Variable	Coefficient Regression b(i)	Erreur standard Sb(i)	Test de T H0: B(i)=0	Prob<	Test de puissance à 5%
Ordonnée	-0.1221	0.0105	-11.599	0.001	1.000

Prélèvement	0.8353	0.0134	62.393	0.001	1.000
St élites	-0.5484	0.0136	-40.272	0.001	1.000
St initiale	-0.007	0.0004	-17.466	0.001	1.000
St élites*St initiale	0.0541	0.0006	95.389	0.001	1.000

Tableau 5. Analyse de variance du modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges élites. PRELEVEMEN=Prélèvement, PW_MARTELA= St élites, ST_INIT=St intital.

Analysis of Variance Section

Source	DF	R2	Sum of Squares	Mean Square	F-Ratio	Prob Level	Power (5%)
Intercept	1		33054.79	33054.79			
Model	4	0.3708	26252.16	6563.041	32878.882	0.0000	1.0000
Error	223144	0.6292	44542.36	0.1996126			
Total(Adjusted)	223148	1.0000	70794.52	0.3172537			

Analysis of Variance Detail Section

Model Term	DF	R2	Sum of Squares	Mean Square	F-Ratio	Prob Level	Power (5%)
Intercept	1		33054.79	33054.79			
Model	4	0.3708	26252.16	6563.041	32878.882	0.0000	1.0000
PRELEVEMEN	1	0.0110	777.0646	777.0646	3892.863	0.0000	1.0000
PW_MARTELA	1	0.0046	323.7339	323.7339	1621.811	0.0000	1.0000
ST_INIT	1	0.0009	60.89513	60.89513	305.067	0.0000	1.0000
PW_MARTELA*ST_INIT	1	0.0257	1816.28	1816.28	9099.021	0.0000	1.0000
Error	223144	0.6292	44542.36	0.1996126			
Total(Adjusted)	223148	1.0000	70794.52	0.3172537			

L'Équation 4 décrit le modèle pour prédire l'indice de changement de différence dans la compétition après traitement pour les tiges élites est donc :

$$\text{Équation 4 : } -.1221 + 0.8353*\text{Prélèvement} - .5484*\text{St élites} - 7.0462\text{E-}03*\text{St initiale} + 5.4063\text{E-}02*\text{St élites}*\text{St initiale}$$

Pour les tiges de bourrage, dans le modèle de régression multiple pour prédire l'indice de changement de différence dans la compétition après traitement, la sélection pas-à-pas montre que les mêmes effets simples sont les plus importants (St en tiges élites, St initiale et prélèvement, ainsi que leurs interactions ont été sélectionnées en premier avant d'inclure les autres variables (DHP de la tige, Différence de DHP moyen après traitement et sa valeur au carré) (Tableau 6). Le modèle final est très similaire à celui sélectionné pour les tiges élites, à la différence près que l'interaction n'est plus entre la surface terrière en tiges élites et la surface terrière initiale mais plutôt entre le prélèvement et la surface terrière en tiges élites.

Tableau 6. Étape de la sélection pas-à-pas des termes du modèle de régression multiple pour les tiges de bourrage

Pas	R ²	Amélioration du R ²	Termes
1	0.028	0.028	St élites
2	0.0345	0.0065	St élites + St initial
3	0.0373	0.0028	St élites + St initial + Prélèvement
4*	0.0435	0.0062	St élites + St initial + Prélèvement + St élites *Prélèvement
5	0.0457	0.0021	St élites + St initial + Prélèvement + St élites *Prélèvement + Delta DHP moyen
6	0.0475	0.0018	St élites + St initial + Prélèvement + St élites *Prélèvement + Delta DHP moyen + Prélèvement * St initial

* Notre modèle final pour les tiges de bourrage est donc :

Équation 5. $ICDCAT^* = St\ élites + St\ initial + Prélèvement + St\ élites *Prélèvement$

*Indice de changement de différence dans la compétition après traitement

Le pouvoir prédictif du modèle pour les tiges de bourrage est beaucoup plus faible ($R^2=0.0435$) que celui pour les tiges élites. La majeure partie de la variance expliquée par notre modèle est aussi celle expliquée par la surface terrière en tiges élites (Tableau 6). Il est intéressant de noter que l'inclusion d'un effet d'interaction entre la surface terrière en tiges élites et le prélèvement a amené à changer le signe de la relation pour les deux effets principaux de cette interaction dans le modèle final (Tableau 7). L'analyse de variance montre les sources d'erreurs du modèle et celles associées à chaque terme (Tableau 8).

Tableau 7. Coefficients des termes utilisés dans le modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges de bourrage

Variable	Coefficient Regression b(i)	Erreur standard Sb(i)	Test de T H0: B(i)=0	Prob<	Test de puissance à 5%
Ordonnée	0.1829	0.0065	28.008	0.001	1.000
Prélèvement	0.2713	0.0165	16.483	0.001	1.000
St élites	0.1216	0.0062	19.457	0.001	1.000
St initiale	-0.0071	0.0002	-41.065	0.001	1.000
St élites*St initiale	-0.8184	0.0203	-40.222	0.001	1.000

Tableau 8. Analyse de variance du modèle final de prédiction de l'indice de changement de différence dans la compétition après traitement pour les tiges de bourrage. PRELEVEMEN=Prélèvement, PW_MARTELA= St élites, ST_INIT=St intital.

Analysis of Variance Section							
Source	DF	R2	Sum of Squares	Mean Square	F-Ratio	Prob Level	Power (5%)
Intercept	1		44.32431	44.32431			
Model	4	0.0435	1325.308	331.3271	2828.531	0.0000	1.0000
Error	248486	0.9565	29107.04	0.1171375			
Total(Adjusted)	248490	1.0000	30432.34	0.1224691			

Analysis of Variance Detail Section							
Term	DF	R2	Sum of Squares	Mean Square	F-Ratio	Prob Level	Power (5%)
Intercept	1		44.32431	44.32431			
Model	4	0.0435	1325.308	331.3271	2828.531	0.0000	1.0000
PRELEVEMEN	1	0.0010	31.8244	31.8244	271.684	0.0000	1.0000
PW_MARTELA	1	0.0015	44.3439	44.3439	378.563	0.0000	1.0000
ST_INIT	1	0.0065	197.5297	197.5297	1686.306	0.0000	1.0000
PRELEVEMEN*PW_MARTELA	1	0.0062	189.5085	189.5085	1617.829	0.0000	1.0000
Error	248486	0.9565	29107.04	0.1171375			
Total(Adjusted)	248490	1.0000	30432.34	0.1224691			

L'Équation 6 décrit donc le modèle pour prédire l'indice de changement de différence dans la compétition après traitement pour les tiges de bourrage.

Équation 6 : $0.1829 + 0.2713 \cdot \text{Prélèvement} + 0.1216 \cdot \text{St élites} - 7.116 \cdot 10^{-3} \cdot \text{St initiale} - 0.8184 \cdot \text{Prélèvement} \cdot \text{St élites}$

Ce sont l'Équation 4 et l'Équation 6 que nous avons intégrées dans *Cohorte* pour les tiges élites et les tiges de bourrage respectivement.

4 – Discussion

Modélisation des effets simples du changement de la différence de compétition après traitement

Nos résultats montrent que la CJ+ permet effectivement de dégager significativement les tiges élites. On observe, en moyenne, peu importe le traitement de CJ+, une réduction significative de l'indice de compétition de $0.385 \text{ m}^2/\text{ha}$. À notre surprise, le fait de concentrer le dégagement autour des tiges élites ne semble pas avoir contribué à augmenter la compétition sur les tiges de bourrage, comme le montre la différence de $0.013 \text{ m}^2/\text{ha}$

(positive!) de changement d'indice de compétition générale de ces tiges. Ce résultat nous a laissé perplexe mais l'analyse plus poussée des résultats a permis d'expliquer ce phénomène. Dans la CJ+, comme les tiges récoltées étaient celles qui compétitionnaient le plus les tiges élites, il s'agissait plus souvent qu'autrement de grosses tiges. Conséquemment, le diamètre moyen (moyenne quadratique) des arbres résiduels après coupe était significativement plus petit que celui avant coupe, indiquant ainsi un prélèvement par le haut. Dans la CJ, le prélèvement demeurait distribué équitablement entre les classes de diamètre, maintenant le diamètre moyen des arbres après coupe similaire à celui avant coupe. Ainsi, dans la CJ+, même les tiges de bourrage se sont trouvées à être plus dégagées (très légèrement) que les arbres résiduels après la CJ. Ce résultat nous a indiqué qu'il serait important d'entrer une variable présentant la différence dans le type de prélèvement (par le haut, équilibré, par le bas), d'où l'ajout de la différence entre la moyenne quadratique du DHP des tiges avant et après traitement entre la CJ et la CJ+ dans notre modèle final.

On note très peu d'effet du DHP de la tige sur l'indice à part peut-être un dégagement moins fort pour les très grosses tiges élites. Ceci est facilement expliqué par le fait que les grosses tiges sont souvent déjà très dégagées et que la CJ+ ne peut apporter que très peu d'augmentation de ce dégagement pour les tiges de ces tailles compte tenu de la formulation de notre indice de compétition.

Nous observons que les effets de la CJ+, soit un dégagement accru des tiges élites accompagné d'une hausse de la compétition pour les tiges de bourrage, seront plus marqués dans les peuplements possédant une plus forte surface terrière initiale et lorsque le taux de prélèvement appliqué dans la CJ+ est plus important. Ces deux résultats combinés démontrent que plus l'espace est limité, plus la gestion des ouvertures prend son importance. Un taux de prélèvement plus grand permet donc plus de latitude pour rediriger les ressources de croissance sur les bonnes tiges d'avenir.

Il apparaît clair que la surface terrière en tiges élites est la variable qui a le plus de pouvoir prédictif sur l'indice de la différence de changement de compétition après traitement. On remarque que l'effet de dégagement accru de la CJ+ décroît rapidement avec l'augmentation de la surface terrière en tiges élites. Ainsi, le dégagement accru sera 6 fois plus important avec 0.5 m²/ha en tiges élites qu'avec 3 m²/ha et plus en tiges élites! C'est à ces faibles surfaces terrières en tiges élites que l'on note aussi la plus grande divergence entre les tiges élites et celles de bourrage, c.-à-d. que la redistribution de l'espace semble la mieux gérée.

L'effet du type de prélèvement, exprimé ici par la différence de moyenne de DHP des arbres après traitement, ne semble pas avoir d'effet sur les tiges élites. Cependant, comme présumé, on observe que les tiges de bourrage obtiennent un indice de changement de la différence de compétition après traitement négatif lorsqu'il n'y a pas de différence en DHP moyen.

La modélisation de l'effet du dégagement accru sur la croissance dans le modèle *Cohorte*

Les deux modèles finaux (Équation 4 et Équation 6) ont été utilisés pour inclure l'effet du jardinage avec dégagement accru des tiges élites dans le modèle *Cohorte* sur la croissance des tiges élites et de bourrage. Pour ce faire, *Cohorte* calcule dans un premier temps la valeur de l'indice de compétition comme s'il s'agissait d'un jardinage avec dégagement balancé entre les tiges. Puis, *Cohorte* identifie les tiges marquées positivement pour un dégagement accru par l'utilisateur dans le module de martelage et calcule la différence de l'indice de compétition à l'aide de l'Équation 4 pour les tiges élites. Pour ce faire, *Cohorte* calcule au préalable les valeurs des variables nécessaires à l'application de la formule, soit la surface terrière initiale, la surface terrière en tiges élites et le taux de prélèvement. Il applique aussi la modification de l'indice de compétition aux tiges de bourrage (celles qui ne sont pas marquées positivement dans le module de martelage) en utilisant l'Équation 6. *Cohorte* simule alors la croissance des arbres selon les équations de croissance usuelles mais en utilisant les indices de compétition modifiés pour le premier quinquennal de simulation après coupe. Pour le deuxième quinquennal (soit de 5 à 10 ans), nous avons supposé que l'effet du dégagement accru serait alors de moitié. Ainsi, le changement dans l'indice de compétition est alors réduit de moitié pour ce deuxième quinquennal. Au troisième quinquennal, nous avons assumé que l'effet de dégagement accru s'est estompé et que les règles normales d'application de la croissance dans le modèle s'applique.

Évaluation de l'effet du dégagement accru sur la croissance dans le modèle *Cohorte*

Il faut néanmoins souligner que la modification de l'indice de compétition demeure, somme toute, assez modeste. À titre d'exemple, avec 0.5 m²/ha en tiges élites, l'indice de compétition est réduit, après CJ+, en général, peu importe le niveau de prélèvement, de 1.2 m²/ha comparativement à après CJ. Une telle réduction en compétition est plutôt faible lorsqu'on la place en contexte des relations entre la croissance des tiges et l'indice de compétition dans *Cohorte*. Ainsi, pour la courbe de croissance de l'érable à sucre dans la version paramétrisée pour le Nouveau-Brunswick de *Cohorte*, cela se traduit par une très légère augmentation de la croissance selon la courbe présentée en Figure 17.

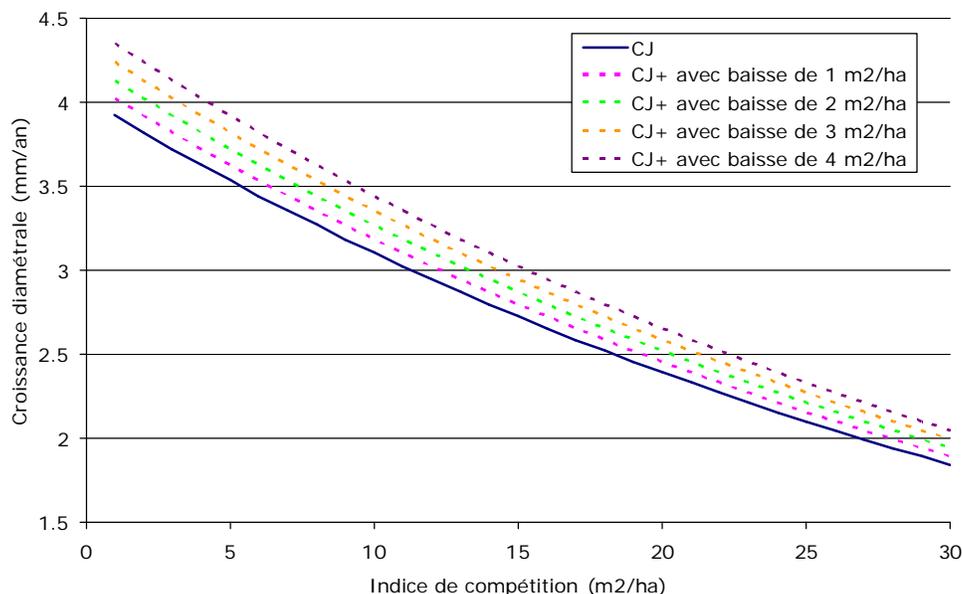


Figure 17. Croissance dans la version paramétrisée pour le Nouveau-Brunswick de *Cohorte* pour des érables à sucre identifiés comme tiges élites en fonction de l'indice de compétition subi par la tige après CJ et CJ+ avec différent niveau de baisse de la compétition pour les tiges vigoureuses.

5 – Conclusion

Le jardinage avec dégagement accru des tiges élites (CJ+) est une nouvelle approche sylvicole pour laquelle il n'existe que très peu de recherches effectuées. L'approche préconisée dans cette étude utilise la modélisation de l'environnement de compétition pour estimer l'effet de la CJ+. Cette approche est innovatrice puisque par ces travaux, nous démontrons qu'il est possible d'inclure des effets d'organisation spatiale des ouvertures dans un modèle de croissance des peuplements qui ne soit pas spatialement explicite. Une telle démonstration est importante puisque la majorité des organisations qui sont responsables de modéliser les rendements des peuplements forestiers utilisent des modèles de simulation par tiges indépendant de la distance, compte tenu de l'effort nécessaire pour la caractérisation spatiale des tiges lors des inventaires.

Les résultats obtenus dans cette étude sont dépendants de la structure des peuplements et de l'organisation des arbres. Nous pensons néanmoins avoir sélectionné une gamme diversifiée d'organisations structurales. Bien que nous n'ayons pas caractérisé la structure

de l'organisation spatiale des arbres, une évaluation visuelle de la localisation des arbres dans les 49 PÉs nous montre que plusieurs patrons spatiaux étaient représentés. Ceci expliquerait la très grande variation de réponse de notre indice de changement de la différence de compétition après traitement.

Le modèle général de prédiction du changement de la différence de la compétition après traitement obtenu est satisfaisant pour les tiges élites mais plutôt faible pour les tiges de bourrage. Il existe une très grande variation de réponse pour les tiges de bourrage. Nous n'avons pas réussi à vraiment déceler un patron de réduction de la compétition après jardinage avec dégagement accru des tiges élites pour les tiges de bourrage. Cela ne veut pas dire qu'il n'y en a pas.

Cette étude nous a permis de distinguer plusieurs éléments importants concernant l'application du jardinage avec dégagement accru des tiges élites (CJ+).

- 1) Nos résultats nous suggèrent qu'il est important de limiter le nombre de tiges élites dans l'application de la CJ+ sinon l'avantage de la sélection sur les tiges d'avenir est réduit considérablement. Par exemple, en sélectionnant de 1-1.5 m²/ha en tiges élites à chaque rotation, nous permettons un dégagement accru de 10 à 15 tiges/ha de 40 cm. Bien que cette densité semble faible, rappelons que ce taux s'applique à chaque rotation et permettra, sur 100 ans, de favoriser la croissance d'une cinquantaine de tiges d'avenir.
- 2) Bien que notre modèle global montre que les tiges de bourrage, pour un même type de prélèvement (c.-à-d. même moyenne de DHP entre la CJ et la CJ+ après traitement), sont plus compétitionnées après CJ+, cet effet demeure très faible. Ce résultat nous étonne car nous avons posé l'hypothèse que le dégagement accru des élites serait obligatoirement accompagné d'une augmentation de la compétition après traitement des tiges de bourrage. Cela est probablement dû au fait que la majeure partie du peuplement demeure composé des tiges de bourrage, répartissant ainsi l'effet de réduction sur un plus grand nombre de tiges que l'effet du dégagement accru sur les tiges élites. Il semble donc qu'il soit possible de faire de la CJ+ sans trop affecter la croissance générale du peuplement. Cela est d'autant plus vrai si le prélèvement s'effectue un peu plus par le haut pour la CJ+ (comme cela risque d'ailleurs d'arriver dans la plupart des cas); en dégageant les tiges d'avenir par le haut, les tiges de bourrage sont d'autant plus dégagées en même temps. Des essais

de rendement de la CJ+ par modélisation seront prochainement réalisés afin de vérifier cette hypothèse.

- 3) L'effet du dégagement accru demeure cependant plutôt modeste. En effet, le dégagement accru pour les tiges élites se traduit par une baisse de l'indice de compétition de 0.2 à 1.2 m²/ha, avec baisse moyenne de 0.385 m²/ha. Comme présenté, lorsque transposé en valeur de croissance dans le modèle *Cohorte*, ces baisses de compétition n'occasionne qu'un changement léger de la croissance. Rappelons néanmoins que ce modèle est la résultante d'une moyenne générale d'une population de simulations qui comporte énormément de variation. Il demeure, d'après nous, encore possible de générer un dégagement accru pour stimuler significativement la croissance des arbres élites. Néanmoins, nos résultats nous montre qu'en régime de coupe partielle, la latitude pour faire cette différence sur la croissance des tiges élites n'est pas aussi grande que nous l'avions présumée au départ de cette recherche. Nous nous demandons si l'application pratique ne se traduira pas en une hausse plus marquée de la croissance. Le dispositif expérimental dans le cadre de ce projet visera justement à vérifier comment la croissance des tiges élites se comporte après CJ+.

6 – Littérature citée

- Bédard, S., Brassard, F. (2002). Les effets réels des coupe de jardinages dans les forêts publiques du Québec en 1995 et 1996. Ministère des ressources naturelles (MRN). Québec, 25p.
- Bédard, S., S. Meunier, L. Blais and Z. Majcen. 2004. Les effets réels des coupes de jardinage dans les forêts publiques du Québec de 1995 à 1998. Rapport technique n° 483. Gouvernement du Québec, Ministère des Ressources naturelles, Direction de la recherche forestière, Sainte-Foy, Québec, Canada. 48 p.
- Bédard, S., Majcen, Z., Meunier, S. 2003. Growth following single-tree selection cutting in Québec Northern Harwoods. *The Forestry Chronicle* 79(5):898-905.
- Doyon, F., P. Nolet et R. Pouliot. 2005. *COHORTE : un modèle de croissance et d'évolution de la qualité adapté à l'application de coupes partielles*. Institut Québécois d'Aménagement de la Forêt Feuillue (IQAFF), Ripon, Québec. Rapport technique, 31 p.

- Doyon F, Nolet P., Forget É. et Pouliot R. 2006. *COHORTE*: a distance-independent individual tree model to assess stand growth and quality grade changes under partial cutting regimes. Proceedings of the IUFRO 1.05 conference on Natural disturbance-based silviculture – Managing for complexity, Rouyn-Noranda, Québec, Canada. p.194
- Forget, E., Nolet, P., Doyon, F., Delagrangé, S., Jardon, Y. (2007). Ten-year response of northern hardwood stands to commercial selection cutting in southern Québec, Canada. *Forest Ecology and Management* 242:764-775.
- Lessard, G., et autres. 2005. Détermination des paramètres des forêts aptes au régime du jardinage (phase I), Sainte-Foy, Centre collégial de transfert de technologie en foresterie, 156 p.
- Majcen, Z. Richard, Y. Ménard, M. et Grenier, Y. 1990. Choix des tiges à marquer pour le jardinage d'érablières inéquiennes – guide technique. Mémoire n°96. Ministère de l'Énergie et Ressources, Gouvernement du Québec, 96 p.
- Majcen, Z., Bédard, S., et Meunier, S. 2005. Accroissement et Mortalité quinze ans après la coupe de jardinage dans quatorze érablières du Québec méridional, MRNF, DRF, Québec, 56p. Majcen 1990
- R Development Core Team . 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. <http://www.R-project.org>.

Annexes

Annexe 1. Code source Visual Basic pour le prélèvement balancé et avec dégagement accru des tiges élites

Code source VB6 Prélèvement balancé

```
Private Sub Command2_Click()
'bouton pour selectionner une base de données access
CDlg1.CancelError = True
'---- Détermine le type de fichiers à sélectionner
CDlg1.Filter = "Fichiers Access " & _
"(*.mdb)|*.mdb"
'---- Affiche la boite de dialogue
CDlg1.ShowOpen
If CDlg1.FileName <> "" Then
    '---- Affiche le nom du fichier sélectionné
    Form1.Text1.Text = CDlg1.FileName
End If
Form1.lst_tbl_form.Clear
Call ouvrir_connection
Call ouvrir_connection_mysql
'conn.Open
Set rs = CONN.OpenSchema(adSchemaTables)
ReDim liste_table(0)
i = 0
While Not rs.EOF
    If rs.Fields("TABLE_TYPE") = "TABLE" Then
        'MsgBox rs!TABLE_NAME
        Form1.lst_tbl_form.AddItem rs!TABLE_NAME
        ReDim Preserve liste_table(UBound(liste_table, 1) + 1)
        liste_table(i) = rs!TABLE_NAME
        i = i + 1
    End If
    rs.MoveNext
Wend
rs.Close
Set rs = Nothing
End Sub
Private Sub lst_tbl_form_Click()
'permet de transféré le contenu dune table dans la seconde liste en cliquant
'sur une table dans la liste de gauche
```

```
Form1.lst_var_form.Clear
NomTable = Form1.lst_tbl_form.Text
'conn.Open
Set rs = CONN.OpenSchema(adSchemaColumns, Array(Empty, Empty, NomTable,
Empty))
While Not rs.EOF
    Form1.lst_var_form.AddItem rs!COLUMN_NAME
    rs.MoveNext
Wend
Form1.tbl_choisi_form.Text = Form1.lst_tbl_form.Text
'conn.Close
rs.Close
Set rs = Nothing
End Sub
Private Sub lst_var_form_Click()
'permet de selectionner la variable dans la tabl qui sert a identifier la parcelle
Form1.ident_form.Text = Form1.lst_var_form.Text
End Sub
Private Sub Command1_Click()
'variable de verification
'strFile = "C:\Documents and Settings\Administrator\Desktop\prelevement
balance\verif.csv"
'Set fs = CreateObject("Scripting.FileSystemObject")
'fs.deletefile strFile
'verif = "no_tuille;nb de tuile non coupe;but en pourcentage;st but de la cellule;nb de
tige ds cellule;id de la tige extraite;buffer;st de la tige;type1;st a repartir total; st
repartir par voisin; nb de voisin;coupe de tout le territoire;id de la tige coupe alea; st de
la tige preleve"
'Call ecrire_verif(verif)
'verif = ""
'effacement et creation des tables pour resultats
aa = Form1.Check1.Value
If Form1.Check1.Value = 0 Then
    conn_mysql.Execute "DROP TABLE IF EXISTS compilation;"
    conn_mysql.Execute "CREATE TABLE compilation (parcelle char(45),
taux_prelevement double, taux_martelage DOUBLE, no_simul integer, ST_initial_tout
DOUBLE, ST_martele_tout DOUBLE, ST_preleve_tout DOUBLE, ST_initial_parcelle
DOUBLE, ST_martele_parcelle DOUBLE, ST_preleve_parcelle DOUBLE);"
```

```

conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC;"
conn_mysql.Execute "CREATE TABLE compilation_IC (no_tige char(20),parcelle
char(45), taux_prelevement double, taux_martelage DOUBLE, no_simul integer, buffer
char(10), compet DOUBLE);"
conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC_avant;"
conn_mysql.Execute "CREATE TABLE compilation_IC_avant (id_tige integer,no_tige
char(20),parcelle char(45), no_simul integer, compet_dhp075 DOUBLE, compet_dhp
DOUBLE, compet DOUBLE );"
conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC_apres;"
conn_mysql.Execute "CREATE TABLE compilation_IC_apres (no_tige char(20),parcelle
char(45), no_simul integer,martelage integer, prelevement double, type_tige char(20),
compet_dhp075 DOUBLE, compet_dhp DOUBLE, compet DOUBLE );"
max_no_simul = 0
Else
Set rs_max_simul = New ADODB.Recordset
rs_max_simul.Open "SELECT max(s.no_simul) FROM compilation as s;", conn_mysql,
adOpenStatic, adLockOptimistic
max_no_simul = rs_max_simul(0)
End If
'demarrage du processus de calcul
compt_prelev = 0
val_pi = Atn(1) * 4
lst_prelev = Array(0.2, 0.25, 0.3, 0.35, 0.4)
lst_martelage = Array(1, 3, 5, 7, 9)
nomtable_output = Array("dhpdhps", "dhp075dhps", "dhpcdhps", "st", "dhpdhps10m",
"dhp075dhps10m", "dhpcdhps10m", "st10m")
'creation d'un array ayant la liste des parcelles
Set rs_lst_parcelle = New ADODB.Recordset
requete = "SELECT " & Form1.tbl_choisi_form.Text & "." & Form1.ident_form.Text & "
FROM " & Form1.tbl_choisi_form.Text & " GROUP BY " & Form1.tbl_choisi_form.Text &
"." & Form1.ident_form.Text & ";"
rs_lst_parcelle.Open requete, CONN, adOpenStatic, adLockOptimistic
rs_lst_parcelle.MoveFirst
rs_lst_parcelle.MoveLast
rs_lst_parcelle.MoveFirst
ReDim lst_parcelle(rs_lst_parcelle.RecordCount - 1)
For i = 0 To rs_lst_parcelle.RecordCount - 1
lst_parcelle(i) = rs_lst_parcelle(0)

```

```

        rs_lst_parcelle.MoveNext
    Next
' fin de la creation de la liste des parcelles
For z = 1 To Form1.Text2.Text 'pour chacune des simulations
    For i = 0 To UBound(lst_parcelle, 1) 'pour chacune des parcelles
        Set rs_rayon = New ADODB.Recordset
        rs_rayon.Open "SELECT tiges.rayon FROM tiges WHERE (((tiges.Parcelle)='" &
lst_parcelle(i) & "')) GROUP BY tiges.rayon;", CONN, adOpenStatic, adLockOptimistic
        arete = Sin(45 * (val_pi / 180)) * rs_rayon(0)
        mult_ha = 10000 / (val_pi * (rs_rayon(0) ^ 2))
        'permet de créer une table ayant seulement les tiges de la parcelles qui sera
actuellement analyser
        parcelle_active = lst_parcelle(i)
        Call ouvrir_table(parcelle_active)
        'creation de la zone de buffer
        Call extract_carre
        Call choix_buffer
        Call ajout_st_tige
        'calcul du IC avant
        Call calcul_IC_avant(lst_parcelle(i), z)
        For n = 0 To UBound(lst_prelev, 1) 'pour chaque taux de prelevement
            Call prelevement(lst_prelev(n))
            Call enregistrer_resultats(0, lst_prelev(n), lst_parcelle(i), z +
max_no_simul)
            Call calcul_IC_apres(0, lst_prelev(n), lst_parcelle(i), z + max_no_simul)
        Next 'pour chaque taux de prelevement

    Next 'pour chacune des parcelles
CONN.Close
Set CONN = Nothing
Call CompactDB(Form1.Text1.Text)
Call ouvrir_connection
Next 'pour chacune des simulations
'rs_max_simul.Close
Set rs_max_simul = Nothing
'rs_lst_parcelle.Close
Set rs_lst_parcelle = Nothing
'rs_rayon.Close

```

```
Set rs_rayon = Nothing
Close #1
MsgBox ("terminé")
End Sub
Public Sub extract_carre()
'creation de 4 carré qui est a linterieur du cercle et qui contient les tiges
'il y a 4 array qui contient les tiges (carre_a, carre_b etc)
Set rs_buff = New ADODB.Recordset
rs_buff.Open "SELECT parcelle_data.* FROM parcelle_data;", CONN, adOpenStatic,
adLockOptimistic
'getrow permet de mettre les données d'une requete dans un array automatiquement
parcelle_orig = rs_buff.GetRows
'donc parcelle_orig est un array contenant toutes les tiges, leur position, le no parcelle,
lessence, le dhp
'parcelle_orig(0,?) = no de la parcelle
'parcelle_orig(1,?) = no de la tige
'parcelle_orig(2,?) = essence
'parcelle_orig(3,?) = dhp
'parcelle_orig(4,?) = position x
'parcelle_orig(5,?) = position y
'creation des 4 array qui contiendront les même colonnes que parcelle_orig
ReDim carre_a(5, 0)
ReDim carre_b(5, 0)
ReDim carre_c(5, 0)
ReDim carre_d(5, 0)
max_array_a = -1
max_array_b = -1
max_array_c = -1
max_array_d = -1
'chaque tige sera placé dans un carré
For i = 0 To UBound(parcelle_orig, 2)
    If parcelle_orig(4, i) > 0 And parcelle_orig(4, i) < arete Then
        If parcelle_orig(5, i) > 0 And parcelle_orig(5, i) < arete Then
            'max_array = UBound(carre_a, 2)
            ReDim Preserve carre_a(5, max_array_a + 1)
            For j = 0 To UBound(parcelle_orig, 1)
                carre_a(j, max_array_a + 1) = parcelle_orig(j, i)
            Next
        End If
    End If
Next
```

```
max_array_a = max_array_a + 1
Elseif parcelle_orig(5, i) < 0 And parcelle_orig(5, i) > -arete Then
'max_array = UBound(carre_b, 2)
ReDim Preserve carre_b(5, max_array_b + 1)
For j = 0 To UBound(parcelle_orig, 1)
    carre_b(j, max_array_b + 1) = parcelle_orig(j, i)
Next
max_array_b = max_array_b + 1
End If

Elseif parcelle_orig(4, i) < 0 And parcelle_orig(4, i) > -arete Then
If parcelle_orig(5, i) > 0 And parcelle_orig(5, i) < arete Then
'max_array = UBound(carre_d, 2)
ReDim Preserve carre_d(5, max_array_d + 1)
For j = 0 To UBound(parcelle_orig, 1)
    carre_d(j, max_array_d + 1) = parcelle_orig(j, i)
Next
max_array_d = max_array_d + 1
Elseif parcelle_orig(5, i) < 0 And parcelle_orig(5, i) > -arete Then
'max_array = UBound(carre_c, 2)
ReDim Preserve carre_c(5, max_array_c + 1)
For j = 0 To UBound(parcelle_orig, 1)
    carre_c(j, max_array_c + 1) = parcelle_orig(j, i)
Next
max_array_c = max_array_c + 1
End If
End If

Next
rs_buff.Close
Set rs_buff = Nothing
End Sub

Public Sub choix_buffer()
taille_array = 0
Call delete_table_if_exist("buffer")
CONN.Execute "CREATE TABLE buffer (Parcelle text(50), NO_TIGE DOUBLE, ESSENCE
text(50), DHP DOUBLE, X DOUBLE, Y DOUBLE, buffer text(45));"
'creation dun tableau avec les nombre 1 à 4 de facon aléatoire
Call nb_aleatoire(1, 4, 4)
Select Case nb_alea(0) 'A
```

```
Case 1
  Call creation_buffer(carre_a, 0, arete, -(2 * arete), (2 * arete), "A")
Case 2
  Call creation_buffer(carre_b, 0, 0, -(2 * arete), (2 * arete), "A")
Case 3
  Call creation_buffer(carre_c, -arete, 0, -(2 * arete), (2 * arete), "A")
Case 4
  Call creation_buffer(carre_d, -arete, arete, -(2 * arete), (2 * arete), "A")
End Select
Select Case nb_alea(1) 'B
  Case 1
    Call creation_buffer(carre_a, 0, arete, -arete, (2 * arete), "B")
  Case 2
    Call creation_buffer(carre_b, 0, 0, -arete, (2 * arete), "B")
  Case 3
    Call creation_buffer(carre_c, -arete, 0, -arete, (2 * arete), "B")
  Case 4
    Call creation_buffer(carre_d, -arete, arete, -arete, (2 * arete), "B")
End Select
Select Case nb_alea(2) 'C
  Case 1
    Call creation_buffer(carre_a, 0, arete, 0, (2 * arete), "C")
  Case 2
    Call creation_buffer(carre_b, 0, 0, 0, (2 * arete), "C")
  Case 3
    Call creation_buffer(carre_c, -arete, 0, 0, (2 * arete), "C")
  Case 4
    Call creation_buffer(carre_d, -arete, arete, 0, (2 * arete), "C")
End Select
Select Case nb_alea(3) 'D
  Case 1
    Call creation_buffer(carre_a, 0, arete, arete, (2 * arete), "D")
  Case 2
    Call creation_buffer(carre_b, 0, 0, arete, (2 * arete), "D")
  Case 3
    Call creation_buffer(carre_c, -arete, 0, arete, (2 * arete), "D")
  Case 4
    Call creation_buffer(carre_d, -arete, arete, arete, (2 * arete), "D")
```

End Select

Call nb_aleatoire(1, 4, 4)

Select Case nb_alea(0) 'E

Case 1

Call creation_buffer(carre_a, 0, arete, arete, arete, "E")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, arete, "E")

Case 3

Call creation_buffer(carre_c, -arete, 0, arete, arete, "E")

Case 4

Call creation_buffer(carre_d, -arete, arete, arete, arete, "E")

End Select

Select Case nb_alea(1) 'F

Case 1

Call creation_buffer(carre_a, 0, arete, arete, 0, "F")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, 0, "F")

Case 3

Call creation_buffer(carre_c, -arete, 0, arete, 0, "F")

Case 4

Call creation_buffer(carre_d, -arete, arete, arete, 0, "F")

End Select

Select Case nb_alea(2) 'G

Case 1

Call creation_buffer(carre_a, 0, arete, arete, -arete, "G")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, -arete, "G")

Case 3

Call creation_buffer(carre_c, -arete, 0, arete, -arete, "G")

Case 4

Call creation_buffer(carre_d, -arete, arete, arete, -arete, "G")

End Select

Select Case nb_alea(3) 'H

Case 1

Call creation_buffer(carre_a, 0, arete, 0, -arete, "H")

Case 2

Call creation_buffer(carre_b, 0, 0, 0, -arete, "H")

Case 3

```
        Call creation_buffer(carre_c, -arete, 0, 0, -arete, "H")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, 0, -arete, "H")
End Select
Call nb_aleatoire(1, 4, 4)
Select Case nb_alea(0) 'I
    Case 1
        Call creation_buffer(carre_a, 0, arete, -arete, -arete, "I")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -arete, -arete, "I")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -arete, -arete, "I")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -arete, -arete, "I")
End Select
Select Case nb_alea(1) 'J
    Case 1
        Call creation_buffer(carre_a, 0, arete, -(2 * arete), -arete, "J")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -(2 * arete), -arete, "J")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -(2 * arete), -arete, "J")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -(2 * arete), -arete, "J")
End Select
Select Case nb_alea(2) 'K
    Case 1
        Call creation_buffer(carre_a, 0, arete, -(2 * arete), 0, "K")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -(2 * arete), 0, "K")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -(2 * arete), 0, "K")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -(2 * arete), 0, "K")
End Select
Select Case nb_alea(3) 'L
    Case 1
        Call creation_buffer(carre_a, 0, arete, -(2 * arete), arete, "L")
```

```

Case 2
    Call creation_buffer(carre_b, 0, 0, -(2 * arete), arete, "L")
Case 3
    Call creation_buffer(carre_c, -arete, 0, -(2 * arete), arete, "L")
Case 4
    Call creation_buffer(carre_d, -arete, arete, -(2 * arete), arete, "L")
End Select
Call delete_table_if_exist("orig_buffer")
requete1 = "SELECT buffer.Parcelle, buffer.NO_TIGE, buffer.ESSENCE, buffer.DHP,
buffer.X, buffer.Y, buffer.buffer INTO orig_buffer FROM buffer;"
CONN.Execute requete1
requete2 = "INSERT INTO orig_buffer ( Parcelle, NO_TIGE, ESSENCE, DHP, X, Y, buffer )
SELECT parcelle_data.Parcelle, parcelle_data.NO_TIGE, parcelle_data.ESSENCE,
parcelle_data.DHP, parcelle_data.X, parcelle_data.Y, 'orig' FROM parcelle_data;"
CONN.Execute requete2
requete3 = "DELETE orig_buffer.buffer, Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2))
AS Expr1 FROM orig_buffer WHERE (((orig_buffer.buffer)<>'orig') AND
((Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2)))<=" & rs_rayon(0) & "));"
    CONN.Execute "DELETE orig_buffer.buffer,
Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2)) AS Expr1 FROM orig_buffer WHERE
(((orig_buffer.buffer)<>'orig') AND
((Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2)))<=" & rs_rayon(0) & "));"
' ajout dune variable contenant un numero unique pour identifier la tige
requete4 = "ALTER TABLE orig_buffer ADD COLUMN id_tige COUNTER CONSTRAINT
MyFieldConstraint PRIMARY KEY;"
CONN.Execute requete4
Call delete_table_if_exist("buffer")
End Sub
Public Sub creation_buffer(aa, refx, refy, diffx, diffy, nom_buffer)
var_x = diffx - refx
var_y = diffy - refy
ajout = aa
For i = 0 To UBound(aa, 2)
    ajout(4, i) = aa(4, i) + var_x
    ajout(5, i) = aa(5, i) + var_y
Next
Set rs_commun = New ADODB.Recordset
rs_commun.Open "SELECT * FROM buffer;", CONN, adOpenDynamic, adLockOptimistic

```

```

For i = 0 To UBound(ajout, 2)
    rs_commun.AddNew
    For j = 0 To UBound(ajout, 1)
        rs_commun(j) = ajout(j, i)
    Next
    rs_commun(6) = nom_buffer
Next
rs_commun.Update
rs_commun.Close
Set rs_commun = Nothing
End Sub

Public Sub ajout_st_tige()
CONN.Execute "ALTER TABLE orig_buffer ADD COLUMN st_tige DOUBLE;"
superficie_total = (arete * 4) ^ 2
CONN.Execute "UPDATE orig_buffer SET st_tige =
(3.141592654*(orig_buffer.DHP/2/100)^2);"
End Sub

Public Sub prelevement(taux_prelev)
'calcul de la ST qu'il me faut preleve
Set rs_st_orig = New ADODB.Recordset
rs_st_orig.Open "SELECT Sum(tbl.st_tige) AS Expr1 FROM orig_buffer as tbl;", CONN,
adOpenStatic, adLockOptimistic
but_st = (rs_st_orig(0) * taux_prelev)
st_preleve = 0
rayon = 10
Set rs_st_arbre_moyen = New ADODB.Recordset
rs_st_arbre_moyen.Open "SELECT Avg(3.141592654*([orig_buffer]![DHP]/2/100)^2)
AS Expr1 FROM orig_buffer;", CONN, adOpenStatic, adLockOptimistic
superficie_total = (arete * 4) ^ 2
but_st = (rs_st_orig(0) * taux_prelev)
rs_st_orig.Close
superficie_tuile_brut = (rs_st_arbre_moyen(0) * superficie_total) / but_st
rs_st_arbre_moyen.Close
nombre_tuile = superficie_total / superficie_tuile_brut
nombre_tuile_range_brut = Sqr(nombre_tuile)
nombre_tuile_range = Round(nombre_tuile_range_brut, 0)
taille_vert_horiz = arete * 4
taille_tuile = taille_vert_horiz / nombre_tuile_range

```

```

nb_tuile_tot = nombre_tuile_range ^ 2
'selection des tiges qui pourront-êre coupé
Call delete_table_if_exist("tige_a_couper")
CONN.Execute "select orig_buffer.* into tige_a_couper from orig_buffer;"
CONN.Execute "ALTER TABLE tige_a_couper ADD COLUMN x_positif DOUBLE, y_positif
DOUBLE, numero_cellule integer;"
CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.x_positif =
[tige_a_couper]![X]+(" & arete & "*"2), tige_a_couper.y_positif = [tige_a_couper]![y]+("
& arete & "*"2);"
CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.numero_cellule =
(Int([tige_a_couper]![x_positif]/" & taille_tuile & "))+(Int([tige_a_couper]![y_positif]/"
& taille_tuile & "))*" & nombre_tuile_range & ");"
'ajout dune collone dans la table tige_a_couper indiquant si la tige est preleve ou non
CONN.Execute "ALTER TABLE tige_a_couper ADD COLUMN tige_preleve text(4);"
CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'non';"
Call delete_table_if_exist("st_cellule")
CONN.Execute "SELECT tige_a_couper.numero_cellule, Sum(tige_a_couper.st_tige) AS
st_cellule INTO st_cellule FROM tige_a_couper GROUP BY
tige_a_couper.numero_cellule;"
CONN.Execute "ALTER TABLE st_cellule ADD COLUMN x_position DOUBLE, y_position
DOUBLE, voisin text(25), but double, coupe_oui_non text(5), st_coupe double;"
CONN.Execute "UPDATE st_cellule SET st_cellule.y_position =
Int([st_cellule]![numero_cellule]/" & nombre_tuile_range & "))*" & taille_tuile & ",
st_cellule.x_position = ([st_cellule]![numero_cellule] mod " & nombre_tuile_range & "))*"
& taille_tuile & ";"
CONN.Execute "UPDATE st_cellule SET st_cellule.but = st_cellule.st_cellule*" &
taux_prelev & ";"
CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'non';"
Call nb_aleatoire(0, Int(nombre_tuile_range ^ 2) - 1, Int(nombre_tuile_range ^ 2))
nb_tuile_noncoupe = nombre_tuile_range ^ 2
'visite de chacune des tuiles
For i = 0 To UBound(nb_alea)
    nb_tuile_noncoupe = nb_tuile_noncoupe - 1
    Set rs_tige_tuile = New ADODB.Recordset
    rs_tige_tuile.Open "SELECT tige_a_couper.* FROM tige_a_couper WHERE
(((tige_a_couper.numero_cellule)=" & nb_alea(i) & ")) AND
((tige_a_couper.tige_preleve)='non') ORDER BY

```

```

Rnd(Now()*tige_a_couper.no_tige),Rnd(Now()*tige_a_couper.x),Rnd(Now()*tige_a_couper.y);", CONN, adOpenStatic, adLockOptimistic
    If rs_tige_tuile.EOF = False Then
        array_tige = rs_tige_tuile.GetRows
        Call atteinte_but(nb_alea(i), nb_tuile_noncoupe, taux_prelev, array_tige)
    End If
    rs_tige_tuile.Close
Next
' une fois toute le tuiles visité, alors faire le fine tuning
recommence:
Set rs_verif_parcelle = New ADODB.Recordset
rs_verif_parcelle.Open "SELECT Sum(tige_a_couper.st_tige) AS SumOfst_tige,
Sum(If([tige_a_couper].[tige_preleve]='oui',[tige_a_couper].[st_tige],0)) AS recolte,
Sum(If(tige_a_couper!tige_preleve='non',tige_a_couper!st_tige,0)) AS residuel,
Sum(If([tige_a_couper].[tige_preleve]='non',1,0)) AS nbtige_res FROM tige_a_couper
WHERE (((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
st_originale = rs_verif_parcelle(0)
st_recolte = rs_verif_parcelle(1)
st_residuel = rs_verif_parcelle(2)
nb_tige_residuel = rs_verif_parcelle(3)
st_attendu = st_originale * taux_prelev
rs_verif_parcelle.Close
If st_recolte < st_attendu Then
    Call nb_aleatoire(0, (nb_tige_residuel - 1), nb_tige_residuel * 1)
    Set rs_tige_tuile2 = New ADODB.Recordset
    rs_tige_tuile2.Open "SELECT tige_a_couper.* FROM tige_a_couper WHERE
(((tige_a_couper.buffer)='orig') AND ((tige_a_couper.tige_preleve)='non'));", CONN,
adOpenStatic, adLockOptimistic
    tige_a_coupe = rs_tige_tuile2.GetRows
    rs_tige_tuile2.Close
    For u = 0 To UBound(nb_alea)
        CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui'
WHERE (((tige_a_couper.NO_TIGE)=" & tige_a_coupe(1, nb_alea(u)) & ") AND
((tige_a_couper.buffer)='orig'));"
        st_recolte = st_recolte + tige_a_coupe(8, nb_alea(u))
        If st_recolte > st_attendu Then
            Exit For
        End If
    End For

```

```

Next
End If
' creation d'une table regroupant la st coupe de chaque tuile
delete_table_if_exist ("st_coupe_par_tuile")
CONN.Execute "SELECT tige_a_couper.numero_cellule,
Sum(IIf(tige_a_couper!tige_preleve='oui',tige_a_couper!st_tige,0)) AS st_coupe INTO
st_coupe_par_tuile FROM tige_a_couper GROUP BY tige_a_couper.numero_cellule
HAVING (((Sum((IIf([tige_a_couper]![buffer]='orig',1,0))))>0)) ORDER BY
Sum(IIf(tige_a_couper!tige_preleve='oui',tige_a_couper!st_tige,0)) DESC;"
Set rs_verif_parcelle = New ADODB.Recordset
rs_verif_parcelle.Open "SELECT Sum(tige_a_couper.st_tige) AS SumOfst_tige,
Sum(IIf([tige_a_couper]![tige_preleve]='oui',[tige_a_couper]![st_tige],0)) AS recolte,
Sum(IIf(tige_a_couper!tige_preleve='non',tige_a_couper!st_tige,0)) AS residuel,
Sum(IIf([tige_a_couper]![tige_preleve]='non',1,0)) AS nbtige_res FROM tige_a_couper
WHERE (((tige_a_couper.buffer)='orig'));" , CONN, adOpenStatic, adLockOptimistic
st_originale_ha = rs_verif_parcelle(0) * mult_ha
st_recolte_ha = rs_verif_parcelle(1) * mult_ha
st_residuel_ha = rs_verif_parcelle(2) * mult_ha
nb_tige_residuel_ha = rs_verif_parcelle(3) * mult_ha
st_attendu_ha = st_originale_ha * taux_prelev
rs_verif_parcelle.Close
If st_recolte_ha > st_attendu_ha + 0.5 Then
    no_ordre_tuile = 0
    Set rs_tuile_st_recolte = New ADODB.Recordset
    rs_tuile_st_recolte.Open "SELECT st_coupe_par_tuile.numero_cellule FROM
st_coupe_par_tuile ORDER BY st_coupe_par_tuile.st_coupe DESC;" , CONN,
adOpenStatic, adLockOptimistic
    tuile_recolte_trie = rs_tuile_st_recolte.GetRows
    rs_tuile_st_recolte.Close
    While st_recolte_ha > st_attendu_ha + 0.5
        id_tuile = tuile_recolte_trie(0, no_ordre_tuile)
        Set rs_petite_tige = New ADODB.Recordset
        rs_petite_tige.Open "SELECT TOP 1 tige_a_couper.st_tige, tige_a_couper.NO_TIGE
FROM tige_a_couper WHERE (((tige_a_couper.numero_cellule)=" & id_tuile & ") AND
((tige_a_couper.tige_preleve)='oui') AND ((tige_a_couper.buffer)='orig')) ORDER BY
tige_a_couper.st_tige;" , CONN, adOpenStatic, adLockOptimistic
        'sassurer qu'il y a au moins une tige non-coupe dans la tuile
        If rs_petite_tige.RecordCount > 0 Then

```

```

    numero_cellule = rs_petite_tige(0)
    st_tige_coupe_cellule = rs_petite_tige(1)
    CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'non'
WHERE (((tige_a_couper.NO_TIGE)=" & rs_petite_tige(1) & ") AND
((tige_a_couper.buffer)='orig'));"
    Else
        numero_cellule = "vide"
        st_tige_coupe_cellule = "vide"
    End If
    Set rs_verif_parcelle1 = New ADODB.Recordset
    rs_verif_parcelle1.Open "SELECT Sum(tige_a_couper.st_tige) AS SumOfst_tige,
Sum(IIf([tige_a_couper]![tige_preleve]='oui',[tige_a_couper]![st_tige],0)) AS recolte,
Sum(IIf(tige_a_couper!tige_preleve='non',tige_a_couper!st_tige,0)) AS residuel,
Sum(IIf([tige_a_couper]![tige_preleve]='non',1,0)) AS nbtige_res FROM tige_a_couper
WHERE (((tige_a_couper.buffer)='orig'));" , CONN, adOpenStatic, adLockOptimistic
    st_recolte_ha = rs_verif_parcelle1(1) * mult_ha
    rs_verif_parcelle1.Close
    'pour recommencer si il a visiter toute les tuiles
    If UBound(tuile_recolte_trie, 2) = no_ordre_tuile Then
        no_ordre_tuile = 0
        Set rs_tuile_st_recolte = New ADODB.Recordset
        rs_tuile_st_recolte.Open "SELECT st_coupe_par_tuile.numero_cellule FROM
st_coupe_par_tuile ORDER BY st_coupe_par_tuile.st_coupe DESC;" , CONN,
adOpenStatic, adLockOptimistic
        tuile_recolte_trie = rs_tuile_st_recolte.GetRows
        rs_tuile_st_recolte.Close
    End If
    no_ordre_tuile = no_ordre_tuile + 1
    rs_petite_tige.Close
Wend
End If
'verification si je ne retombe pas en dessous de ma limite et verification pour ne pas
tomber dans une boucle sans fin
If st_recolte_ha < st_attendu_ha And boucle < 20 Then
    'on recommence
    boucle = boucle + 1
    GoTo recommence
End If

```

```

Set rs_st_orig = Nothing
Set rs_st_arbre_moyen = Nothing
Set rs_calcul_voisin1 = Nothing
Set rs_calcul_voisin2 = Nothing
Set rs_nb_tuile_plein = Nothing
Set rs_tige_tuile = Nothing
Set rs_verif_parcelle = Nothing
Set rs_tige_a_coupe = Nothing
Set rs_verif_parcelle = Nothing
Set rs_tuile_st_recolte = Nothing
Set rs_petite_tige = Nothing
Set rs_tige_tuile2 = Nothing
End Sub

Public Sub calcul_IC_avant(parcelle, no_simul)
Set rs_liste_tige = New ADODB.Recordset
rs_liste_tige.Open "SELECT orig_buffer.id_TIGE, orig_buffer.DHP, orig_buffer.X,
orig_buffer.Y, orig_buffer.buffer, orig_buffer.no_tige FROM orig_buffer;", CONN,
adOpenStatic, adLockOptimistic
liste_tige = rs_liste_tige.GetRows
'tige_martelee(0,..) = ID de la tige
'tige_martelee(1,..) = DHP de la tige
'tige_martelee(2,..) = position X de la tige
'tige_martelee(3,..) = position Y de la tige
'tige_martelee(4,..) = buffer de la tige
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_avant1;"
conn_mysql.Execute "CREATE TABLE table_IC_avant1 (no_tige_sujet
char(20),no_tige_comp char(20),compet DOUBLE);"
superficie_total = (val_pi * (rs_rayon(0) ^ 2))
'condition de competition: 1-distance entre les tige <10 2-seul lestiges originale sont
utilisé comme tige sujet 3- une meme tige ne peut etre sa propre competitrice
' 4- dhpsujet*0.75 <= dhpcompetitrice
For x = 0 To UBound(liste_tige, 2)
    If liste_tige(4, x) = "orig" Then
        conn_mysql.Execute "INSERT INTO table_IC_avant1 ( no_tige_sujet ,no_tige_comp
,compet ) SELECT "" & liste_tige(0, x) & "" , "",0 ;"
        For y = 0 To UBound(liste_tige, 2)

```

```

        If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3, y))
^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) = "orig"
And liste_tige(1, x) * 0.75 <= liste_tige(1, y) Then
        st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 / superficie_total
        conn_mysql.Execute "INSERT INTO table_IC_avant1 ( no_tige_sujet
,no_tige_comp ,compet ) SELECT "" & liste_tige(0, x) & "" , "" & liste_tige(0, y) & "" , "" &
st_tige & "" ;"
        End If
    Next
End If
Next
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_avant2;"
conn_mysql.Execute "CREATE TABLE table_IC_avant2 (no_tige_sujet
char(20),no_tige_comp char(20),compet DOUBLE);"
superficie_total = (val_pi * (rs_rayon(0) ^ 2))
'condition de competition: 1-distance entre les tige <10 2-seul lestiges originale sont
utilisé comme tige sujet 3- une meme tige ne peut etre sa propre competitrice
' 4- dhpsujet < dhpcompetitrice
For x = 0 To UBound(liste_tige, 2)
    If liste_tige(4, x) = "orig" Then
        conn_mysql.Execute "INSERT INTO table_IC_avant2 ( no_tige_sujet ,no_tige_comp
,compet ) SELECT "" & liste_tige(0, x) & "" , "",0 ;"
        For y = 0 To UBound(liste_tige, 2)
            If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3,
y)) ^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) =
"orig" And liste_tige(1, x) <= liste_tige(1, y) Then
                st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 /
superficie_total
                conn_mysql.Execute "INSERT INTO table_IC_avant2 ( no_tige_sujet
,no_tige_comp ,compet ) SELECT "" & liste_tige(0, x) & "" , "" & liste_tige(0, y) & "" , "" &
st_tige & "" ;"
            End If
        Next
    End If
Next
End If
Next
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_avant3;"
conn_mysql.Execute "CREATE TABLE table_IC_avant3 (no_tige_sujet
char(20),no_tige_comp char(20),compet DOUBLE);"

```

```

superficie_total = (val_pi * (rs_rayon(0) ^ 2))
'condition de competition: 1-distance entre les tige <10 2-seul lestiges originale sont
utilisé comme tige sujet 3- une meme tige ne peut etre sa propre competitorice
' 4- toute les tiges dans un rayon de 10m
For x = 0 To UBound(liste_tige, 2)
    If liste_tige(4, x) = "orig" Then
        conn_mysql.Execute "INSERT INTO table_IC_avant3 ( no_tige_sujet ,no_tige_comp
,compet ) SELECT "" & liste_tige(0, x) & "" , "",0 ;"
        For y = 0 To UBound(liste_tige, 2)
            If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3,
y)) ^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) =
"orig" Then
                st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 /
superficie_total
                conn_mysql.Execute "INSERT INTO table_IC_avant3 ( no_tige_sujet
,no_tige_comp ,compet ) SELECT "" & liste_tige(0, x) & "" , "" & liste_tige(0, y) & "" , "" &
st_tige & "" ;"
            End If
        Next
    End If
Next
Next
For x = 1 To 3 'pour chacun des trois indice
    conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_avant_compile" & x & ";"
    conn_mysql.Execute "create table table_IC_avant_compile" & x & " SELECT
t.no_tige_sujet, sum(t.compet) as compet FROM ccfb_nb2.table_ic_avant" & x & " as t
group by t.no_tige_sujet;"
Next
conn_mysql.Execute "INSERT INTO compilation_IC_avant ( id_tige, compet_dhp075,
compet_dhp,compet, no_simul, parcelle ) SELECT
table_ic_avant_compile3.no_tige_sujet, table_ic_avant_compile1.compet,
table_ic_avant_compile2.compet, table_ic_avant_compile3.compet, "" & no_simul & "" , ""
& parcelle & "" & "" FROM table_ic_avant_compile1 RIGHT JOIN table_ic_avant_compile3
ON table_ic_avant_compile1.no_tige_sujet = table_ic_avant_compile3.no_tige_sujet
LEFT JOIN table_ic_avant_compile2 ON table_ic_avant_compile3.no_tige_sujet =
table_ic_avant_compile2.no_tige_sujet;"
'ajout du numero de tige originale
For x = 0 To UBound(liste_tige, 2)

```

```

conn_mysql.Execute "UPDATE compilation_ic_avant SET compilation_ic_avant.no_tige
= " & liste_tige(5, x) & " WHERE (((compilation_ic_avant.id_tige)=" & liste_tige(0, x) &
"));";
Next
rs_liste_tige.Close
Set rs_liste_tige = Nothing
End Sub

Public Sub calcul_IC_apres(martelage, prelev, parcelle, no_simul)
Set rs_liste_tige = New ADODB.Recordset
rs_liste_tige.Open "SELECT tige_a_couper.id_tige, tige_a_couper.DHP, tige_a_couper.X,
tige_a_couper.Y, tige_a_couper.buffer, tige_a_couper.tige_preleve,
tige_a_couper.no_tige FROM tige_a_couper where tige_a_couper.tige_preleve = 'non';",
CONN, adOpenStatic, adLockOptimistic
liste_tige = rs_liste_tige.GetRows
'tige_martele(0,..) = ID de la tige
'tige_martele(1,..) = DHP de la tige
'tige_martele(2,..) = position X de la tige
'tige_martele(3,..) = position Y de la tige
'tige_martele(4,..) = buffer de la tige
'tige_martele(5,..) = preleve oui/non
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_apres1;"
conn_mysql.Execute "CREATE TABLE table_IC_apres1 (no_tige_sujet
text(20),no_tige_comp text(20),compet DOUBLE,no_tige integer);"
superficie_total = (val_pi * (rs_rayon(0) ^ 2))
For x = 0 To UBound(liste_tige, 2)
  If liste_tige(4, x) = "orig" Then
    conn_mysql.Execute "INSERT INTO table_IC_apres1 ( no_tige_sujet ,no_tige_comp
,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , ',0,' & liste_tige(6, x) & '";"
    For y = 0 To UBound(liste_tige, 2)
      If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3, y))
^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) = "orig"
And liste_tige(1, x) * 0.75 <= liste_tige(1, y) Then
        st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 / superficie_total
        conn_mysql.Execute "INSERT INTO table_IC_apres1 ( no_tige_sujet
,no_tige_comp ,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , '" & liste_tige(0, y) &
'",' & st_tige & ',' & liste_tige(6, x) & '";"
      End If
    Next
  Next
Next

```

```

End If
Next
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_apres2;"
conn_mysql.Execute "CREATE TABLE table_IC_apres2 (no_tige_sujet
text(20),no_tige_comp text(20),compet DOUBLE,no_tige integer);"
superficie_total = (val_pi * (rs_rayon(0) ^ 2))
For x = 0 To UBound(liste_tige, 2)
  If liste_tige(4, x) = "orig" Then
    conn_mysql.Execute "INSERT INTO table_IC_apres1 ( no_tige_sujet ,no_tige_comp
,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , ",0," & liste_tige(6, x) & ";"
    For y = 0 To UBound(liste_tige, 2)
      If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3, y))
^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) = "orig"
And liste_tige(1, x) <= liste_tige(1, y) Then
        st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 / superficie_total
        conn_mysql.Execute "INSERT INTO table_IC_apres2 ( no_tige_sujet
,no_tige_comp ,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , '" & liste_tige(0, y)
& "'," & st_tige & "'," & liste_tige(6, x) & ";"
      End If
    Next
  End If
Next
End If
Next
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_apres3;"
conn_mysql.Execute "CREATE TABLE table_IC_apres3 (no_tige_sujet
text(20),no_tige_comp text(20),compet DOUBLE,no_tige integer);"
superficie_total = (val_pi * (rs_rayon(0) ^ 2))
For x = 0 To UBound(liste_tige, 2)
  If liste_tige(4, x) = "orig" Then
    conn_mysql.Execute "INSERT INTO table_IC_apres1 ( no_tige_sujet ,no_tige_comp
,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , ",0," & liste_tige(6, x) & ";"
    For y = 0 To UBound(liste_tige, 2)
      If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3, y))
^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(4, x) = "orig"
Then
        st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 / superficie_total
        conn_mysql.Execute "INSERT INTO table_IC_apres3 ( no_tige_sujet
,no_tige_comp ,compet, no_tige ) SELECT '" & liste_tige(0, x) & "' , '" & liste_tige(0, y)
& "'," & st_tige & "'," & liste_tige(6, x) & ";"
      End If
    Next
  End If
Next

```

```

        End If
    Next
End If
Next
For x = 1 To 3 'pour chacun des trois indice
    conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_apres_compile" & x & ";"
    conn_mysql.Execute "create table table_IC_apres_compile" & x & " SELECT t.no_tige,
sum(t.compet) as compet FROM ccfb_nb2.table_ic_apres" & x & " as t group by
t.no_tige_sujet;"
Next
conn_mysql.Execute "INSERT INTO compilation_IC_apres ( no_tige, parcelle, no_simul,
martelage, prelevement, type_tige, compet_dhp075, compet_dhp,compet ) SELECT
table_ic_apres_compile3.no_tige, "" & parcelle & "" , "" & no_simul & "" , "" & martelage & "" ,
"" & prelev & "" , "" , table_ic_apres_compile1.compet, table_ic_apres_compile2.compet,
table_ic_apres_compile3.compet FROM table_ic_a pres_compile1 RIGHT JOIN
table_ic_apres_compile3 ON table_ic_apres_compile1.no_tige =
table_ic_apres_compile3.no_tige LEFT JOIN table_ic_apres_compile2 ON
table_ic_apres_compile3.no_tige = table_ic_apres_compile2.no_tige;"
'ajout du numero de tige originale
rs_liste_tige.Close
Set rs_liste_tige = New ADODB.Recordset
End Sub
Public Sub enregistrer_resultats(tx_martelage, tx_preleve, parcelle, no_simul)
superficie_total = (arete * 4) ^ 2
Set rs_st_init_tout = New ADODB.Recordset
rs_st_init_tout.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2)) AS
st FROM tige_a_couper;", CONN, adOpenStatic, adLockOptimistic
initial_tout = rs_st_init_tout(0) * 10000 / superficie_total
st_martele_tout = 0
Set rs_st_preleve_tout = New ADODB.Recordset
rs_st_preleve_tout.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2))
AS st FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui'));", CONN,
adOpenStatic, adLockOptimistic
st_preleve_tout = rs_st_preleve_tout(0) * 10000 / superficie_total
Set rs_st_init_parcelle = New ADODB.Recordset
rs_st_init_parcelle.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2))
AS st FROM tige_a_couper WHERE (((tige_a_couper.buffer)='orig'));", CONN,
adOpenStatic, adLockOptimistic

```

```

initial_parcelle = rs_st_init_parcelle(0) * mult_ha
st_martele_parcelle = 0
Set rs_st_preleve_parcelle = New ADODB.Recordset
rs_st_preleve_parcelle.Open "SELECT
Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2)) AS st FROM tige_a_couper WHERE
(((tige_a_couper.tige_preleve)='oui') AND ((tige_a_couper.buffer)='orig'))";, CONN,
adOpenStatic, adLockOptimistic
st_preleve_parcelle = rs_st_preleve_parcelle(0) * mult_ha
conn_mysql.Execute "INSERT INTO compilation ( parcelle, taux_prelevement,
taux_martelage, no_simul, ST_initial_tout, ST_martele_tout, ST_preleve_tout,
ST_initial_parcelle, ST_martele_parcelle, ST_preleve_parcelle ) SELECT "" & parcelle & ""
, "" & tx_preleve & "" , "" & tx_martelage & "" , "" & no_simul & "" , "" & initial_tout & "" , "" &
st_martele_tout & "" , "" & st_preleve_tout & "" , "" & initial_parcelle & "" , "" &
st_martele_parcelle & "" , "" & st_preleve_parcelle & "" ;"
End Sub
Private Sub atteinte_but(no_tuille, nb_tuille_noncoupe, pourc_but, tbl_tige1)
verif1 = no_tuille & ";" & nb_tuille_noncoupe & ";" & pourc_but
ordre_prelev = 1
Set rs_but_st = New ADODB.Recordset
rs_but_st.Open "SELECT st_cellule.but FROM st_cellule WHERE
(((st_cellule.numero_cellule)=" & no_tuille & "));", CONN, adOpenStatic,
adLockOptimistic
but_st = rs_but_st(0)
' condition si le but de recolte de cette tige est negatif, alors il ne faut pas coupé et sortir
immédiatement
If but_st < 0 Then
    CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'oui' WHERE
((st_cellule.numero_cellule)=" & no_tuille & " ) ;"
    Exit Sub
End If
verif1 = verif1 & ";" & but_st & ";" & UBound(tbl_tige1, 2) + 1
If UBound(tbl_tige1, 2) = 0 Then
    st_enleve = tbl_tige1(8, 0)
    CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui' WHERE
(((tige_a_couper.NO_TIGE)=" & tbl_tige1(1, tige_extraite) & ") AND
((tige_a_couper.buffer)="" & tbl_tige1(6, tige_extraite) & ""));"
    verif = verif1 & ";" & (tbl_tige1(1, 0)) & ";" & (tbl_tige1(6, 0)) & ";" & (tbl_tige1(8,
0)) & ";" ;"

```



```

End If
'a la fin de la visite de la tuile je repartie ce qui a ete coupe en trop ou ce qui manque à
coupé
Set rs_st_recolte = New ADODB.Recordset
rs_st_recolte.Open "SELECT Sum(tige_a_couper.st_tige) AS SumOfst_tige FROM
tige_a_couper WHERE (((tige_a_couper.numero_cellule)=" & no_tuile & ") AND
((tige_a_couper.tige_preleve)='oui'))";", CONN, adOpenStatic, adLockOptimistic
If IsNull(rs_st_recolte(0)) Then st_recolte = 0 Else st_recolte = rs_st_recolte(0)
st_a_repartir = but_st - st_recolte
If nb_tuile_noncoupe = 0 Then
    st_reparti_final = st_a_repartir
Else
    st_a_repartir_voisin = st_a_repartir / nb_tuile_noncoupe
    CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'oui' WHERE
((st_cellule.numero_cellule)=" & no_tuile & ") ;"
    CONN.Execute "UPDATE st_cellule SET st_cellule.but = [st_cellule].[but]+(" &
st_a_repartir_voisin & ") WHERE (((st_cellule.coupe_oui_non)='non'));"
End If
verif = verif1 & ";;;" & st_a_repartir & ";" & st_a_repartir_voisin
verif = ""
End Sub

```

Modules vb6

```

Type Tableau
    ligne() As String
    col() As String
End Type
Public rs_rayon As ADODB.Recordset
Public CONN As ADODB.Connection
Public conn_mysql As ADODB.Connection
Public rs As ADODB.Recordset
Public parcelle_data() As String
Public nb_alea() As String
Public nb_alea2() As String
Public parcelle_all()
Public ajout
Public taille_array
Public carre_a
Public carre_b

```

```
Public carre_c
Public carre_d
Public arete
Public FN_database
Public val_pi
Public numero_simul
Public liste_table
Public nomtable_output
Public niveau_prelev
Public parcelle_recolte
Public mult_ha
Public compt_prelev
Public Ist_prelev
Public Ist_parcelle
Public array_tuile
Public parcelle_active
Public nb_tuile_tot
Public pourc_but
Public Sub ouvrir_connection()
FN_database = Form1.Text1.Text
Set CONN = New ADODB.Connection
Provider = "Microsoft.JET.OLEDB.4.0"
Connexion = FN_database
CONN.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source= " &
Connexion
CONN.Open
End Sub
Public Sub ouvrir_connection_mysql()
    Set conn_mysql = New ADODB.Connection

    conn_mysql.CursorLocation = adUseClient
    conn_mysql.ConnectionString = "DRIVER={MySQL ODBC 3.51 Driver};" _
        & "SERVER=localhost;" _
        & "DATABASE=ccfb_nb2;" _
        & "UID=root;" _
        & "PWD=vi0I0n;" _
```

```

& "OPTION=" & 1 + 2 + 8 + 32 + 2048 + 16384

conn_mysql.Open
End Sub
Public Sub delete_table_if_exist(table As String)
Set rs = CONN.OpenSchema(adSchemaTables)
While Not rs.EOF
    If rs.Fields("TABLE_TYPE") = "TABLE" And rs!TABLE_NAME = table Then
        CONN.Execute "drop table " & table
    End If
    rs.MoveNext
Wend
rs.Close
End Sub
Public Sub delete_table_if_exist_mysql(table As String)
Set rs = conn_mysql.OpenSchema(adSchemaTables)
While Not rs.EOF
    If rs.Fields("TABLE_TYPE") = "TABLE" And rs!TABLE_NAME = table Then
        conn_mysql.Execute "drop table " & table
    End If
    rs.MoveNext
Wend
rs.Close
End Sub
Public Sub ouvrir_table(condition)
nom_table = Form1.tbl_choisi_form.Text
Call delete_table_if_exist("parcelle_data")
'requete = "INSERT INTO parcelle_data ( Parcelle, NO_TIGE, ESSENCE , DHP, X, Y )
SELECT " & nom_table & ".Parcelle, " & nom_table & ".NO_TIGE, " & nom_table &
".ESSENCE, " & nom_table & ".DHP, " & nom_table & ".X, " & nom_table & ".Y FROM " &
nom_table & " WHERE (((" & nom_table & ".Parcelle)=" & condition & "));"
requete = "SELECT " & nom_table & ".Parcelle, " & nom_table & ".NO_TIGE, " &
nom_table & ".ESSENCE, " & nom_table & ".DHP, " & nom_table & ".X, " & nom_table &
".Y INTO parcelle_data FROM " & nom_table & " WHERE (((" & nom_table & ".Parcelle)=" &
condition & "));"
'debug.print requete
CONN.Execute requete
End Sub

```

```

Public Sub nb_aleatoire(min As Integer, max As Integer, nombre As Integer)
Randomize Timer
ReDim nb_alea(nombre - 1)
i = 0
While i < nombre
  presence = False
  valeur_aleat = Int(Rnd() * (max - min + 1)) + min
  For j = 0 To UBound(nb_alea)
    If nb_alea(j) = "" Then
      ElseIf nb_alea(j) = valeur_aleat Then
        presence = True
      End If
    Next
  If presence = False Then
    nb_alea(i) = valeur_aleat
    i = i + 1
  End If
Wend
End Sub

Public Function aleatoire(min, max)
Randomize Timer
aleatoire = Int(Rnd() * (max - min + 1)) + min
'return valeur
End Function

Public Sub ecrire_verif(ligne)
chemindataexport = "C:\Documents and Settings\Administrator\Desktop\prelevement
balance\verif.csv"
Open chemindataexport For Output As #1
  'print #1, ligne
End Sub

Public Function CompactDB(pFileName As String) As Boolean
'-----
' PROCEDURE : CompactDB
' DESCRIPTION :
' RETURN VALUE: true or false depending wether there were errors or not
' NOTE :
' Add "Microsoft Jet and Replication Objects 2.x Library" to
' the references of your project

```

' EXAMPLE :

```
'-----  
On Error GoTo ErrH  
Dim CONN As New JRO.JetEngine  
Dim ConnstringSorg As String, ConnstringDest As String  
' Ensure file is not read only  
SetAttr pFileName, vbNormal  
ConnstringSorg = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _  
pFileName & ";User ID=;Password=;"  
ConnstringDest = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _  
App.Path & "\Temp.mdb" & ";Jet OLEDB:Engine Type=5;"  
Screen.MousePointer = vbHourglass  
CONN.CompactDatabase ConnstringSorg, ConnstringDest  
Screen.MousePointer = vbDefault  
'Copia il file compattato.  
Kill pFileName  
FileCopy App.Path & "\Temp.mdb", pFileName  
Kill App.Path & "\Temp.mdb"  
Set CONN = Nothing  
CompactDB = True  
Exit Function  
ErrH:  
Screen.MousePointer = vbDefault  
'debug.print Err.Description  
End Function
```

Code source VB6 Prélèvement débalancé

```
Private Sub Command2_Click()
'bouton pour sélectionner une base de données access
CDlg1.CancelError = True
'---- Détermine le type de fichiers à sélectionner
CDlg1.Filter = "Fichiers Access " & _
"(*.mdb)|*.mdb"
'---- Affiche la boîte de dialogue
CDlg1.ShowOpen
If CDlg1.FileName <> "" Then
    '---- Affiche le nom du fichier sélectionné
    Form1.Text1.Text = CDlg1.FileName
End If
Form1.lst_tbl_form.Clear
Call ouvrir_connection
Call ouvrir_connection_mysql
Set rs = CONN.OpenSchema(adSchemaTables)
ReDim liste_table(0)
i = 0
While Not rs.EOF
    If rs.Fields("TABLE_TYPE") = "TABLE" Then

        Form1.lst_tbl_form.AddItem rs!TABLE_NAME
        ReDim Preserve liste_table(UBound(liste_table, 1) + 1)
        liste_table(i) = rs!TABLE_NAME
        i = i + 1
    End If
    rs.MoveNext
Wend
End Sub
Private Sub lst_tbl_form_Click()
'permet de transféré le contenu dune table dans la seconde liste en cliquant
'sur une table dans la liste de gauche
Form1.lst_var_form.Clear
NomTable = Form1.lst_tbl_form.Text
Set rs = CONN.OpenSchema(adSchemaColumns, Array(Empty, Empty, NomTable,
Empty))
While Not rs.EOF
```

```
Form1.lst_var_form.AddItem rs!COLUMN_NAME
rs.MoveNext
Wend
Form1.tbl_choisi_form.Text = Form1.lst_tbl_form.Text
End Sub
Private Sub lst_var_form_Click()
'permet de selectionner la variable dans la tabl qui sert a identifier la parcelle
Form1.ident_form.Text = Form1.lst_var_form.Text
End Sub
Private Sub Command1_Click()
'effacement et creation des tables pour resultats
aa = Form1.Check1.Value
If Form1.Check1.Value = 0 Then
    conn_mysql.Execute "DROP TABLE IF EXISTS compilation;"
    conn_mysql.Execute "CREATE TABLE compilation (parcelle char(45),
taux_prelevement double, taux_martelage DOUBLE, no_simul integer, ST_initial_tout
DOUBLE, ST_martele_tout DOUBLE, ST_preleve_tout DOUBLE, ST_initial_parcelle
DOUBLE, ST_martele_parcelle DOUBLE, ST_preleve_parcelle DOUBLE);"
    conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC;"
    conn_mysql.Execute "CREATE TABLE compilation_IC (no_tige char(20),parcelle
char(45), taux_prelevement double, taux_martelage DOUBLE, no_simul integer, buffer
char(10), compet DOUBLE);"
    conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC_avant;"
    conn_mysql.Execute "CREATE TABLE compilation_IC_avant (no_tige char(20),parcelle
char(45), no_simul integer, compet DOUBLE);"
    conn_mysql.Execute "DROP TABLE IF EXISTS compilation_IC_apres;"
    conn_mysql.Execute "CREATE TABLE compilation_IC_apres (no_tige char(20),parcelle
char(45), no_simul integer,martelage integer, prelevement double, type_tige char(20),
compet DOUBLE);"
    max_no_simul = 0
Else
    Set rs_max_simul = New ADODB.Recordset
    rs_max_simul.Open "SELECT max(s.no_simul) FROM compilation as s;", conn_mysql,
adOpenStatic, adLockOptimistic
    If IsNull(rs_max_simul(0)) Then
        max_no_simul = 0
    Else
        max_no_simul = rs_max_simul(0)
    End If
End If
End Sub
```

```

End If
rs_max_simul.Close
Set rs_max_simul = Nothing
End If
'demarrage du processus de calcul
compt_prelev = 0
val_pi = Atn(1) * 4
lst_prelev = Array(0.2, 0.25, 0.3, 0.35, 0.4)
lst_martelage = Array(1, 3, 5, 7, 9)
nomtable_output = Array("dhpdhps", "dhp075dhps", "dhpcdhps", "st", "dhpdhps10m",
"dhp075dhps10m", "dhpcdhps10m", "st10m")
'creation d'un array ayant la liste des parcelles
Set rs_lst_parcelle = New ADODB.Recordset
requete = "SELECT " & Form1.tbl_choisi_form.Text & "." & Form1.ident_form.Text & "
FROM " & Form1.tbl_choisi_form.Text & " GROUP BY " & Form1.tbl_choisi_form.Text &
"." & Form1.ident_form.Text & ";"
rs_lst_parcelle.Open requete, CONN, adOpenStatic, adLockOptimistic
rs_lst_parcelle.MoveFirst
rs_lst_parcelle.MoveLast
rs_lst_parcelle.MoveFirst
ReDim lst_parcelle(rs_lst_parcelle.RecordCount - 1)
For i = 0 To rs_lst_parcelle.RecordCount - 1
    lst_parcelle(i) = rs_lst_parcelle(0)
    rs_lst_parcelle.MoveNext
Next
rs_lst_parcelle.Close
Set rs_lst_parcelle = Nothing
' fin de la creation de la liste des parcelles
For z = 1 To Form1.Text2.Text 'pour chacune des simulations
    For i = 0 To UBound(lst_parcelle, 1) 'pour chacune des parcelles
        Set rs_rayon = New ADODB.Recordset
        rs_rayon.Open "SELECT tiges.rayon FROM tiges WHERE (((tiges.Parcelle)='" &
lst_parcelle(i) & "')) GROUP BY tiges.rayon;", CONN, adOpenStatic, adLockOptimistic
        arete = Sin(45 * (val_pi / 180)) * rs_rayon(0)
        mult_ha = 10000 / (val_pi * (rs_rayon(0) ^ 2))
        'permet de créer une table ayant seulement les tiges de la parcelles qui sera
actuellement analyser
        parcelle_active = lst_parcelle(i)
    
```

```

Call ouvrir_table(parcelle_active)
'creation de la zone de buffer
Call extract_carre
Call choix_buffer
Call ajout_st_tige
'calcul du IC avant
Call calcul_IC_avant(Ist_parcelle(i), z)
For m = 0 To UBound(Ist_martelage, 1) 'pour chacun de taux de martelage
positif
    Call choix_tige_martele_positif(Ist_martelage(m))
    For n = 0 To UBound(Ist_prelev, 1) 'pour chaque taux de prelevement
        Call prelevement(Ist_prelev(n))
        Call enregistrer_resultats(Ist_martelage(m), Ist_prelev(n),
Ist_parcelle(i), z + max_no_simul)
        Call calcul_IC_apres(Ist_martelage(m), Ist_prelev(n), Ist_parcelle(i), z +
max_no_simul)
    Next 'pour chaque taux de prelevement
Next 'pour chacun de taux de martelage positif

'
Next 'pour chacune des simulations
Next 'pour chacune des parcelles
CONN.Close
Set CONN = Nothing
Call CompactDB(Form1.Text1.Text)
Call ouvrir_connection
Next 'pour chacune des simulations
MsgBox ("terminé")
End Sub
Public Sub extract_carre()
'creation de 4 carré qui est a linterieur du cercle et qui contient les tiges
'il y a 4 array qui contient les tiges (carre_a, carre_b etc)
Set rs_buff = New ADODB.Recordset
rs_buff.Open "SELECT parcelle_data.* FROM parcelle_data;", CONN, adOpenStatic,
adLockOptimistic
'getrow permet de mettre les données d'une requete dans un array automatiquement
parcelle_orig = rs_buff.GetRows
rs_buff.Close
Set rs_buff = Nothing

```

'donc parcelle_orig est un array contenant toutes les tiges, leur position, le no parcelle, l'essence, le dhp

'parcelle_orig(0,?) = no de la parcelle

'parcelle_orig(1,?) = no de la tige

'parcelle_orig(2,?) = essence

'parcelle_orig(3,?) = dhp

'parcelle_orig(4,?) = position x

'parcelle_orig(5,?) = position y

'creation des 4 array qui contiendront les même colonnes que parcelle_orig

ReDim carre_a(5, 0)

ReDim carre_b(5, 0)

ReDim carre_c(5, 0)

ReDim carre_d(5, 0)

max_array_a = -1

max_array_b = -1

max_array_c = -1

max_array_d = -1

'chaque tige sera placé dans un carré

For i = 0 To UBound(parcelle_orig, 2)

 If parcelle_orig(4, i) > 0 And parcelle_orig(4, i) < arete Then

 If parcelle_orig(5, i) > 0 And parcelle_orig(5, i) < arete Then

 ReDim Preserve carre_a(5, max_array_a + 1)

 For j = 0 To UBound(parcelle_orig, 1)

 carre_a(j, max_array_a + 1) = parcelle_orig(j, i)

 Next

 max_array_a = max_array_a + 1

 ElseIf parcelle_orig(5, i) < 0 And parcelle_orig(5, i) > -arete Then

 ReDim Preserve carre_b(5, max_array_b + 1)

 For j = 0 To UBound(parcelle_orig, 1)

 carre_b(j, max_array_b + 1) = parcelle_orig(j, i)

 Next

 max_array_b = max_array_b + 1

 End If

 ElseIf parcelle_orig(4, i) < 0 And parcelle_orig(4, i) > -arete Then

 If parcelle_orig(5, i) > 0 And parcelle_orig(5, i) < arete Then

 ReDim Preserve carre_d(5, max_array_d + 1)

 For j = 0 To UBound(parcelle_orig, 1)

 carre_d(j, max_array_d + 1) = parcelle_orig(j, i)

```
Next
max_array_d = max_array_d + 1
Elseif parcelle_orig(5, i) < 0 And parcelle_orig(5, i) > -arete Then
ReDim Preserve carre_c(5, max_array_c + 1)
For j = 0 To UBound(parcelle_orig, 1)
    carre_c(j, max_array_c + 1) = parcelle_orig(j, i)
Next
max_array_c = max_array_c + 1
End If
End If
Next
End Sub
Public Sub choix_buffer()
taille_array = 0
Call delete_table_if_exist("buffer")
CONN.Execute "CREATE TABLE buffer (Parcelle text(50), NO_TIGE DOUBLE, ESSENCE
text(50), DHP DOUBLE, X DOUBLE, Y DOUBLE, buffer text(45));"
'creation dun tableau avec les nombre 1 à 4 de facon aléatoire
Call nb_aleatoire(1, 4, 4)
Select Case nb_alea(0) 'A
    Case 1
        Call creation_buffer(carre_a, 0, arete, -(2 * arete), (2 * arete), "A")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -(2 * arete), (2 * arete), "A")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -(2 * arete), (2 * arete), "A")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -(2 * arete), (2 * arete), "A")
End Select
Select Case nb_alea(1) 'B
    Case 1
        Call creation_buffer(carre_a, 0, arete, -arete, (2 * arete), "B")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -arete, (2 * arete), "B")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -arete, (2 * arete), "B")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -arete, (2 * arete), "B")
```

End Select

Select Case nb_alea(2) 'C

Case 1

Call creation_buffer(carre_a, 0, arete, 0, (2 * arete), "C")

Case 2

Call creation_buffer(carre_b, 0, 0, 0, (2 * arete), "C")

Case 3

Call creation_buffer(carre_c, -arete, 0, 0, (2 * arete), "C")

Case 4

Call creation_buffer(carre_d, -arete, arete, 0, (2 * arete), "C")

End Select

Select Case nb_alea(3) 'D

Case 1

Call creation_buffer(carre_a, 0, arete, arete, (2 * arete), "D")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, (2 * arete), "D")

Case 3

Call creation_buffer(carre_c, -arete, 0, arete, (2 * arete), "D")

Case 4

Call creation_buffer(carre_d, -arete, arete, arete, (2 * arete), "D")

End Select

Call nb_aleatoire(1, 4, 4)

Select Case nb_alea(0) 'E

Case 1

Call creation_buffer(carre_a, 0, arete, arete, arete, "E")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, arete, "E")

Case 3

Call creation_buffer(carre_c, -arete, 0, arete, arete, "E")

Case 4

Call creation_buffer(carre_d, -arete, arete, arete, arete, "E")

End Select

Select Case nb_alea(1) 'F

Case 1

Call creation_buffer(carre_a, 0, arete, arete, 0, "F")

Case 2

Call creation_buffer(carre_b, 0, 0, arete, 0, "F")

Case 3

```
        Call creation_buffer(carre_c, -arete, 0, arete, 0, "F")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, arete, 0, "F")
End Select
Select Case nb_alea(2) 'G
    Case 1
        Call creation_buffer(carre_a, 0, arete, arete, -arete, "G")
    Case 2
        Call creation_buffer(carre_b, 0, 0, arete, -arete, "G")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, arete, -arete, "G")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, arete, -arete, "G")
End Select
Select Case nb_alea(3) 'H
    Case 1
        Call creation_buffer(carre_a, 0, arete, 0, -arete, "H")
    Case 2
        Call creation_buffer(carre_b, 0, 0, 0, -arete, "H")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, 0, -arete, "H")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, 0, -arete, "H")
End Select
Call nb_aleatoire(1, 4, 4)
Select Case nb_alea(0) 'I
    Case 1
        Call creation_buffer(carre_a, 0, arete, -arete, -arete, "I")
    Case 2
        Call creation_buffer(carre_b, 0, 0, -arete, -arete, "I")
    Case 3
        Call creation_buffer(carre_c, -arete, 0, -arete, -arete, "I")
    Case 4
        Call creation_buffer(carre_d, -arete, arete, -arete, -arete, "I")
End Select
Select Case nb_alea(1) 'J
    Case 1
        Call creation_buffer(carre_a, 0, arete, -(2 * arete), -arete, "J")
```

```

Case 2
  Call creation_buffer(carre_b, 0, 0, -(2 * arete), -arete, "J")
Case 3
  Call creation_buffer(carre_c, -arete, 0, -(2 * arete), -arete, "J")
Case 4
  Call creation_buffer(carre_d, -arete, arete, -(2 * arete), -arete, "J")
End Select
Select Case nb_alea(2) 'K
  Case 1
    Call creation_buffer(carre_a, 0, arete, -(2 * arete), 0, "K")
  Case 2
    Call creation_buffer(carre_b, 0, 0, -(2 * arete), 0, "K")
  Case 3
    Call creation_buffer(carre_c, -arete, 0, -(2 * arete), 0, "K")
  Case 4
    Call creation_buffer(carre_d, -arete, arete, -(2 * arete), 0, "K")
End Select
Select Case nb_alea(3) 'L
  Case 1
    Call creation_buffer(carre_a, 0, arete, -(2 * arete), arete, "L")
  Case 2
    Call creation_buffer(carre_b, 0, 0, -(2 * arete), arete, "L")
  Case 3
    Call creation_buffer(carre_c, -arete, 0, -(2 * arete), arete, "L")
  Case 4
    Call creation_buffer(carre_d, -arete, arete, -(2 * arete), arete, "L")
End Select
Call delete_table_if_exist("orig_buffer")
requete1 = "SELECT buffer.Parcelle, buffer.NO_TIGE, buffer.ESSENCE, buffer.DHP,
buffer.X, buffer.Y, buffer.buffer INTO orig_buffer FROM buffer;"
CONN.Execute requete1
requete2 = "INSERT INTO orig_buffer ( Parcelle, NO_TIGE, ESSENCE, DHP, X, Y, buffer )
SELECT parcelle_data.Parcelle, parcelle_data.NO_TIGE, parcelle_data.ESSENCE,
parcelle_data.DHP, parcelle_data.X, parcelle_data.Y, 'orig' FROM parcelle_data;"
CONN.Execute requete2
requete3 = "DELETE orig_buffer.buffer, Sqr((([orig_buffer]![X]^2)+([orig_buffer]![Y]^2))
AS Expr1 FROM orig_buffer WHERE (((orig_buffer.buffer) <> 'orig') AND
((Sqr((([orig_buffer]![X]^2)+([orig_buffer]![Y]^2))))<=" & rs_rayon(0) & "));"

```

```

CONN.Execute "DELETE orig_buffer.buffer,
Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2)) AS Expr1 FROM orig_buffer WHERE
(((orig_buffer.buffer)<>'orig') AND
((Sqr(([orig_buffer]![X]^2)+([orig_buffer]![Y]^2)))<=" & rs_rayon(0) & "));"
Call delete_table_if_exist("buffer")
End Sub

Public Sub creation_buffer(aa, refx, refy, diffx, diffy, nom_buffer)
var_x = diffx - refx
var_y = diffy - refy
ajout = aa
For i = 0 To UBound(aa, 2)
ajout(4, i) = aa(4, i) + var_x
ajout(5, i) = aa(5, i) + var_y
Next
Set rs_commun = New ADODB.Recordset
rs_commun.Open "SELECT * FROM buffer;", CONN, adOpenDynamic, adLockOptimistic
For i = 0 To UBound(ajout, 2)
rs_commun.AddNew
For j = 0 To UBound(ajout, 1)
rs_commun(j) = ajout(j, i)
Next
rs_commun(6) = nom_buffer
Next
rs_commun.Update
rs_commun.Close
Set rs_commun = Nothing
End Sub

Public Sub ajout_st_tige()
CONN.Execute "ALTER TABLE orig_buffer ADD COLUMN st_tige_ha DOUBLE;"
superficie_total = (arete * 4) ^ 2
CONN.Execute "UPDATE orig_buffer SET st_tige_ha =
(3.141592654*(orig_buffer.DHP/2/100)^2)*10000/" & superficie_total & ";"
End Sub

Public Sub choix_tige_marteles_positif(st_marteles_positif)
On Error Resume Next
Set rs_st_arbre_moyen = New ADODB.Recordset
'calcul de la st moyen des tige <40cm

```

```

rs_st_arbre_moyen.Open "SELECT Avg(3.141592654*([orig_buffer]![DHP]/2/100)^2)
AS Expr1 FROM orig_buffer where [orig_buffer]![DHP]<40 ;", CONN, adOpenStatic,
adLockOptimistic
superficie_total = (arete * 4) ^ 2
st_martele_but = (st_martele_positif * superficie_total) / 10000
'permet de savoir quel serai la taille des cellules si on espère avoir 1 seule tige martele
par cellules
superficie_cellule_brut = (rs_st_arbre_moyen(0) * superficie_total) / st_martele_but
rs_st_arbre_moyen.Close
Set rs_st_arbre_moyen = Nothing
'permet de savoir quel serai le mnombre de cellules si on espère avoir 1 seule tige
martele par cellules
nombre_cellule = superficie_total / superficie_cellule_brut
'arrondir le nombre de cellule (dans une dimension) afin de former un carré avec des
cellules carrés
nombre_cellule_range = Round((Sqr(nombre_cellule)), 0)
taille_vert_horiz = arete * 4
taille_cellule = taille_vert_horiz / nombre_cellule_range
nb_cellule_tot = nombre_cellule_range ^ 2
'creation de la table de taravil contenant les tiges et la cellules
Call delete_table_if_exist("cellules_martelage")
CONN.Execute "select orig_buffer.* into cellules_martelage from orig_buffer where
orig_buffer.dhp < 40;"
CONN.Execute "ALTER TABLE cellules_martelage ADD COLUMN x_positif DOUBLE,
y_positif DOUBLE, numero_cellule integer,classe_dhp text(12), martele text (4);"
CONN.Execute "ALTER TABLE cellules_martelage ADD COLUMN id_tige COUNTER
CONSTRAINT MyFieldConstraint PRIMARY KEY;"
CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele = 'non' ;"
CONN.Execute "UPDATE cellules_martelage SET cellules_ martelage.x_positif =
[cellules_martelage]![X]+(" & arete & "*2), cellules_martelage.y_positif =
[cellules_martelage]![y]+(" & arete & "*2);"
CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.numero_cellule =
int(cellules_martelage.x_positif/" & taille_cellule & ")+(int(cellules_martelage.y_positif/"
& taille_cellule & ")*" & nombre_cellule_range & ");"
CONN.Execute "ALTER TABLE cellules_martelage ADD COLUMN st_tige_ha_orig
DOUBLE;"
CONN.Execute "UPDATE cellules_martelage SET st_tige_ha_orig =
(3.141592654*(cellules_martelage.DHP/2/100)^2)* " & mult_ha & ";"

```

```

Call delete_table_if_exist("st_cellule")
CONN.Execute "SELECT cellules_martelage.numero_cellule,
Sum(cellules_martelage.st_tige_ha) AS st_cellule INTO st_cellule FROM
cellules_martelage where cellules_martelage.dhp < 40 GROUP BY
cellules_martelage.numero_cellule ;"
CONN.Execute "ALTER TABLE st_cellule ADD COLUMN x_position DOUBLE, y_position
DOUBLE, voisin text(25), but double, coupe_oui_non text(5), st_coupe double;"
CONN.Execute "UPDATE st_cellule SET st_cellule.y_position =
Int([st_cellule]![numero_cellule]/" & nombre_cellule_range & ")*" & taille_cellule & ",
st_cellule.x_position = ([st_cellule]![numero_cellule] mod " & nombre_cellule_range &
")*" & taille_cellule & ";"
CONN.Execute "UPDATE st_cellule SET st_cellule.but = " & st_marteles_positif & "/" &
nb_cellule_tot & ";"
CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'non';"
'preparation de certaine variable
'ajout dune variable contenant les classe de dhp
CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.classe_dhp =
Switch([cellules_martelage]![DHP]<20,'10-20',[cellules_martelage]![DHP]<30,'20-
30',[cellules_martelage]![DHP]<40,'30-40',[cellules_martelage]![DHP]>=40,'40');"
Call nb_aleatoire(0, Int(nombre_cellule_range ^ 2) - 1, Int(nombre_cellule_range ^ 2))
nb_cellule_noncoupe = nombre_cellule_range ^ 2
'visite de chacune des tuiles
For i = 0 To UBound(nb_alea)
    nb_cellule_noncoupe = nb_cellule_noncoupe - 1
    Set rs_tige_cellule = New ADODB.Recordset
    rs_tige_cellule.Open "SELECT cellules_martelage.* FROM cellules_martelage WHERE
(((cellules_martelage.numero_cellule)=" & nb_alea(i) & ")) AND
((cellules_martelage.marteles)='non') ORDER BY
Rnd(Now()*cellules_martelage.no_tige),Rnd(Now()*cellules_martelage.x),Rnd(Now()*cel
lules_martelage.y);", CONN, adOpenStatic, adLockOptimistic
    If rs_tige_cellule.EOF = False Then
        array_tige = rs_tige_cellule.GetRows
        rs_tige_cellule.Close
        Call atteinte_but_martelage(nb_alea(i), nb_cellule_noncoupe, st_marteles_positif,
array_tige)
    End If
Next

```

```
If rs_tige_cellule.State = adStateOpen Then
rs_tige_cellule.Close
End If
Set rs_tige_cellule = Nothing
'certaine parcelle n'ont pas assez de tiges pour martelé voir parcelle 0_2 il n'y as que
7m2 de tige en bas de 40 cm
'donc je vais creer un condition que si la st des 40 et moins est inferieur au martelage, il
faut marteler toute les tiges et sortir de cette section
Set rs_st_tige_40 = New ADODB.Recordset
rs_st_tige_40.Open "SELECT Sum(cellules_martelage.st_tige_ha_orig) AS
SumOfst_tige_ha_orig FROM cellules_martelage WHERE
(((cellules_martelage.buffer)='orig'))";, CONN, adOpenStatic, adLockOptimistic
If rs_st_tige_40(0) < st_martele_positif Then
    CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele = 'oui'
WHERE (((cellules_martelage.buffer)='orig'))";
    Exit Sub
End If
' une fois toute le tuilles visité, alors faire le fine tuning
recommence:
Set rs_verif_parcelle = New ADODB.Recordset
rs_verif_parcelle.Open "SELECT Sum(cellules_martelage.st_tige_ha_orig) AS
SumOfst_tige,
Sum(IIf([cellules_martelage]![martele]='oui',[cellules_martelage]![st_tige_ha_orig],0))
AS martele,
Sum(IIf(cellules_martelage!martele='non',cellules_martelage!st_tige_ha_orig,0)) AS
residuel, Sum(IIf([cellules_martelage]![martele]='non',1,0)) AS nbtige_res FROM
cellules_martelage WHERE (((cellules_martelage.buffer)='orig'))";, CONN, adOpenStatic,
adLockOptimistic
st_originale = rs_verif_parcelle(0)
st_recolte = rs_verif_parcelle(1)
st_residuel = rs_verif_parcelle(2)
nb_tige_residuel = rs_verif_parcelle(3)
st_attendu = st_martele_positif
rs_verif_parcelle.Close
If st_recolte < st_attendu Then
    Set rs_tige_tuille2 = New ADODB.Recordset
```

```

rs_tige_tuille2.Open "SELECT cellules_martelage.* FROM cellules_martelage WHERE
(((cellules_martelage.buffer)='orig') AND ((cellules_martelage.martele)='non')) ORDER
BY Rnd(Now()*cellules_martelage.no_tige);", CONN, adOpenStatic, adLockOptimistic
tige_a_coupe = rs_tige_tuille2.GetRows
rs_tige_tuille2.Close
classe_dhp = ""

liste_cond() = Array("", "")
For u = 0 To UBound(tige_a_coupe, 2)
    tige_classe_dhp = tige_a_coupe(11, u)
    If Not tige_classe_dhp = liste_cond(0) Or Not tige_classe_dhp = liste_cond(1) Then
        Call condition_classe_all(liste_cond, tige_classe_dhp)

        CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele =
'oui' WHERE (((cellules_martelage.NO_TIGE)=" & tige_a_coupe(1, u) & ") AND
((cellules_martelage.buffer)='orig'));"
        st_recolte = st_recolte + tige_a_coupe(14, u)
        'apres avoir martele une tige et que lon veut en martele une autre aller voir la
fonction qui permet de choisir les classes de dhp
        If st_recolte > st_attendu Then
            Exit For
        ElseIf u = UBound(tige_a_coupe, 2) Then
            u = -1
            Set rs_tige_tuille3 = New ADODB.Recordset
            rs_tige_tuille3.Open "SELECT cellules_martelage.* FROM cellules_martelage
WHERE (((cellules_martelage.buffer)='orig') AND ((cellules_martelage.martele)='non'))
ORDER BY Rnd(Now()*cellules_martelage.no_tige);", CONN, adOpenStatic,
adLockOptimistic
            tige_a_coupe = rs_tige_tuille3.GetRows
            rs_tige_tuille3.Close
        End If
    End If
Next
End If
' creation d'une table regroupant la st coupe de chaque tuile
delete_table_if_exist ("st_martele_par_tuille")
CONN.Execute "SELECT cellules_martelage.numero_cellule,
Sum(If(cellules_martelage!martele='oui',cellules_martelage!st_tige_ha_orig,0)) AS

```

```

st_marteles INTO st_marteles_par_tuile FROM cellules_martelage GROUP BY
cellules_martelage.numero_cellule HAVING
(((Sum((IIf([cellules_martelage].[buffer]='orig',1,0))))>0)) ORDER BY
Sum(IIf(cellules_martelage.marteles='oui',cellules_martelage.st_tige_ha_orig,0)) DESC;"
Set rs_verif_parcelle = New ADODB.Recordset
rs_verif_parcelle.Open "SELECT Sum(cellules_martelage.st_tige_ha_orig) AS
SumOfst_tige,
Sum(IIf([cellules_martelage].[marteles]='oui',[cellules_martelage].[st_tige_ha_orig],0))
AS marteles,
Sum(IIf(cellules_martelage.marteles='non',cellules_martelage.st_tige_ha_orig,0)) AS
residuel, Sum(IIf([cellules_martelage].[marteles]='non',1,0)) AS nbtige_res FROM
cellules_martelage WHERE (((cellules_martelage.buffer)='orig'));" , CONN, adOpenStatic,
adLockOptimistic
st_originale_ha = rs_verif_parcelle(0)
st_recolte_ha = rs_verif_parcelle(1)
st_residuel_ha = rs_verif_parcelle(2)
nb_tige_residuel_ha = rs_verif_parcelle(3)
st_attendu_ha = st_marteles_positif
rs_verif_parcelle.Close
Set rs_verif_parcelle = Nothing
If st_recolte_ha > st_attendu_ha + 0.5 Then
    no_ordre_tuile = 0
    Set rs_tuile_st_recolte = New ADODB.Recordset
    rs_tuile_st_recolte.Open "SELECT st_marteles_par_tuile.numero_cellule FROM
st_marteles_par_tuile ORDER BY st_marteles_par_tuile.st_marteles DESC;" , CONN,
adOpenStatic, adLockOptimistic
    tuile_recolte_trie = rs_tuile_st_recolte.GetRows
    rs_tuile_st_recolte.Close
    Set rs_verif_parcelle = Nothing
    While st_recolte_ha > st_attendu_ha + 0.5
        id_tuile = tuile_recolte_trie(0, no_ordre_tuile)
        Set rs_petite_tige = New ADODB.Recordset
        rs_petite_tige.Open "SELECT TOP 1 cellules_martelage.st_tige_ha_orig,
cellules_martelage.NO_TIGE FROM cellules_martelage WHERE
(((cellules_martelage.numero_cellule)=" & id_tuile & ") AND
((cellules_martelage.marteles)='oui') AND ((cellules_martelage.buffer)='orig')) ORDER BY
cellules_martelage.st_tige_ha_orig;" , CONN, adOpenStatic, adLockOptimistic
        'sassurer qu'il y a au moins une tige non-coupe dans la tuile

```

```

If rs_petite_tige.RecordCount > 0 Then
    numero_cellule = rs_petite_tige(0)
    st_tige_coupe_cellule = rs_petite_tige(1)
    CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele =
'non' WHERE (((cellules_martelage.NO_TIGE)=" & rs_petite_tige(1) & ") AND
((cellules_martelage.buffer)='orig'));"
    Else
        numero_cellule = "vide"
        st_tige_coupe_cellule = "vide"
    End If
    Set rs_verif_parcelle1 = New ADODB.Recordset
    rs_verif_parcelle1.Open "SELECT Sum(cellules_martelage.st_tige_ha_orig) AS
SumOfst_tige,
Sum(IIf([cellules_martelage]![martele]='oui',[cellules_martelage]![st_tige_ha_orig],0))
AS recolte,
Sum(IIf(cellules_martelage!martele='non',cellules_martelage!st_tige_ha_orig,0)) AS
residuel, Sum(IIf([cellules_martelage]![martele]='non',1,0)) AS nbtige_res FROM
cellules_martelage WHERE (((cellules_martelage.buffer)='orig'));" , CONN, adOpenStatic,
adLockOptimistic
    st_recolte_ha = rs_verif_parcelle1(1)
    rs_verif_parcelle1.Close
    'pour recommencer si il a visiter toute les tuiles
    If UBound(tuile_recolte_trie, 2) = no_ordre_tuile Then
        no_ordre_tuile = 0
        Set rs_tuile_st_recolte = New ADODB.Recordset
        rs_tuile_st_recolte.Open "SELECT st_martele_par_tuile.numero_cellule FROM
st_martele_par_tuile ORDER BY st_martele_par_tuile.st_martele DESC;" , CONN,
adOpenStatic, adLockOptimistic
        tuile_recolte_trie = rs_tuile_st_recolte.GetRows
        rs_tuile_st_recolte.Close
    End If
    no_ordre_tuile = no_ordre_tuile + 1
    rs_petite_tige.Close
    Set rs_petite_tige = Nothing
Wend
End If
'verification si je ne retombe pas en dessous de ma limite et verification pour ne pas
tomber dans une boucle sans fin

```

```
If st_recolte_ha < st_attendu_ha And boucle < 20 Then
    'on recommence
    boucle = boucle + 1
    GoTo recommence
End If
Set rs_st_orig = Nothing
Set rs_calcul_voisin1 = Nothing
Set rs_calcul_voisin2 = Nothing
Set rs_nb_tuile_plein = Nothing
Set rs_tige_tuile = Nothing
Set rs_verif_parcelle = Nothing
Set rs_tige_a_coupe = Nothing
Set rs_verif_parcelle = Nothing
Set rs_tuile_st_recolte = Nothing
Set rs_tige_tuile2 = Nothing
End Sub
Public Sub prelevement(taux_prelev)
On Error Resume Next
'calcul de la ST qu'il me faut preleve
recommence:
Set rs_st_orig = New ADODB.Recordset
rs_st_orig.Open "SELECT Sum(tbl.st_tige_ha) AS Expr1 FROM cellules_martelage as
tbl;", CONN, adOpenStatic, adLockOptimistic
but_st = (rs_st_orig(0) * taux_prelev)
st_preleve = 0
rayon = 10
rs_st_orig.Close
'selection des tiges qui pourront-êre coupé
Call delete_table_if_exist("tige_a_couper")
CONN.Execute "select cellules_martelage.* into tige_a_couper from cellules_martelage;"
'puisue dans la section precedente j'ai utilisé les tiges de 40cm et moin maintenant il
faute que je les rajoute
CONN.Execute "INSERT INTO tige_a_couper ( numero_cellule, classe_dhp, martele )
SELECT orig_buffer.*, 0 AS Expr1, '40+' AS Expr2, 'non' AS Expr3 FROM orig_buffer
WHERE (((orig_buffer.DHP)>=40));"
CONN.Execute "UPDATE tige_a_couper SET st_tige_ha_orig =
(3.141592654*(tige_a_couper.DHP/2/100)^2)* " & mult_ha & ";"
'ajout dune collone dans la table tige_a_couper indiquant si la tige est preleve ou non
```

```

CONN.Execute "ALTER TABLE tige_a_couper ADD COLUMN tige_preleve text(4);"
CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'non';"
terminer = False
aa:
Call delete_table_if_exist("table_choix_ic")
CONN.Execute "CREATE TABLE table_choix_ic (tige_martele int,dhp_tige_mart
double,st_tige_mart double, tige_preleve int,dhp_tige_prelev double,st_tige_prelev
double, distance_tiges double, IC double, preleve text(5), st_tige_ha double);"
Set rs_tige_martele = New ADODB.Recordset
rs_tige_martele.Open "SELECT tige_a_couper.id_TIGE, tige_a_couper.DHP,
tige_a_couper.X, tige_a_couper.Y, tige_a_couper.buffer FROM tige_a_couper where
tige_a_couper.martele = 'oui' ORDER BY Rnd(Now()*tige_a_couper.NO_TIGE);", CONN,
adOpenStatic, adLockOptimistic
Set rs_tige_a_coupe = New ADODB.Recordset
rs_tige_a_coupe.Open "SELECT tige_a_couper.id_TIGE, tige_a_couper.DHP,
tige_a_couper.X, tige_a_couper.Y, tige_a_couper.buffer FROM tige_a_couper where
tige_a_couper.tige_preleve = 'non' and tige_a_couper.martele = 'non';", CONN,
adOpenStatic, adLockOptimistic
tige_martele = rs_tige_martele.GetRows
'tige_martele(0,..) = ID de la tige
'tige_martele(1,..) = DHP de la tige
'tige_martele(2,..) = position X de la tige
'tige_martele(3,..) = position Y de la tige
'tige_martele(4,..) = buffer de la tige
tige_a_coupe = rs_tige_a_coupe.GetRows
'tige_a_coupe(0,..) = ID de la tige
'tige_a_coupe(1,..) = DHP de la tige
'tige_a_coupe(2,..) = position X de la tige
'tige_a_coupe(3,..) = position Y de la tige
'tige_a_coupe(4,..) = buffer de la tige
rs_tige_martele.Close
rs_tige_a_coupe.Close
Set rs_tige_martele = Nothing
Set rs_tige_a_coupe = Nothing
For x = 0 To UBound(tige_martele, 2)
    For y = 0 To UBound(tige_a_coupe, 2)
        distance_tige = ((tige_martele(2, x) - tige_a_coupe(2, y)) ^ 2 + (tige_martele(3,
x) - tige_a_coupe(3, y)) ^ 2) ^ (1 / 2)

```

```

    If distance_tige < 10 Then
        no_tige_marteles = tige_marteles(0, x)
        no_tige_a_coupe = tige_a_coupe(0, y)
        CONN.Execute "INSERT INTO table_choix_ic (
tige_marteles,dhp_tige_mart,tige_preleve,dhp_tige_prelev,distance_tiges ) SELECT "" &
no_tige_marteles & "" , "" & tige_marteles(1, x) & "" , "" & no_tige_a_coupe & "" , "" &
tige_a_coupe(1, y) & "" , "" & distance_tige & "" ;"
    End If
Next
Next
CONN.Execute "UPDATE table_choix_ic SET IC =
table_choix_ic.dhp_tige_prelev/(table_choix_ic.dhp_tige_mart*table_choix_ic.distance_tig
es);"
CONN.Execute "UPDATE table_choix_ic SET preleve = 'non';"
superficie_total = (arete * 4) ^ 2
CONN.Execute "UPDATE table_choix_ic SET st_tige_ha = (" & val_pi &
"*(table_choix_ic.dhp_tige_prelev/2/100)^2)*10000/" & superficie_total & "";"
Set rs_count_tige_a_preleve = New ADODB.Recordset
rs_count_tige_a_preleve.Open "SELECT Count(table_choix_ic.tige_marteles) AS
CompteDetige_marteles FROM table_choix_ic;", CONN, adOpenStatic, adLockOptimistic
count_tige_a_preleve = rs_count_tige_a_preleve(0)
rs_count_tige_a_preleve.Close
Set rs_count_tige_a_preleve = Nothing
'selection de la tige compétitrice ayant la plus grosse IC pour chacune des marteles+
Set rs_tige_a_preleve = New ADODB.Recordset
rs_tige_a_preleve.Open "SELECT table_choix_ic.tige_marteles AS ID_M,
Max(table_choix_ic.IC) AS IC, First(table_choix_ic.dhp_tige_prelev) AS DHP_P,
First(table_choix_ic.tige_preleve) AS ID_P, First(table_choix_ic.st_tige_ha) AS st_P
FROM table_choix_ic WHERE (((table_choix_ic.preleve)='non')) GROUP BY
table_choix_ic.tige_marteles ORDER BY Rnd(Now()*table_choix_ic.tige_marteles);", CONN,
adOpenStatic, adLockOptimistic
If rs_tige_a_preleve.EOF Then
Exit Sub
End If

liste_tige_a_preleve = rs_tige_a_preleve.GetRows
rs_tige_a_preleve.Close
Set rs_tige_a_preleve = Nothing

```

```

st_tige_preleve = 0
For x = 0 To UBound(liste_tige_a_preleve, 2)
    'faire requete qui calcule la st preleve
    Set somme_st_preleve = New ADODB.Recordset
    somme_st_preleve.Open "SELECT Sum([tige_a_couper].[st_tige_ha]) AS st_ha FROM
tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui'))";, CONN, adOpenStatic,
adLockOptimistic
    st_preleve_tot = somme_st_preleve(0)
    st_tige_preleve = liste_tige_a_preleve(4, x)
    no_tige = liste_tige_a_preleve(3, x)
    st_apres = st_preleve_tot + st_tige_preleve
    st_avant = st_preleve_tot
    somme_st_preleve.Close
    Set somme_st_preleve = Nothing
    If st_apres > but_st Then
        If Abs(but_st - st_apres) < Abs(but_st - st_avant) Then
            'si diff de st avant est plus eleve que apres alors ont rajoute cette nouvelle tige
            et on termine la boucle
            CONN.Execute "UPDATE table_choix_ic SET table_choix_ic.preleve = 'oui'
WHERE ((table_choix_ic.tige_preleve)=' & no_tige & ');"
            CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui'
WHERE ((tige_a_couper.id_TIGE)=' & no_tige & ');"
            terminer = True
            Exit For
        Else
            'si diff st_avant est plus petit que diff apres alors on ne rajoute pas la nouvelle
tige et on termine la boucle
            terminer = True
            Exit For
        End If
    Else
        CONN.Execute "UPDATE table_choix_ic SET table_choix_ic.preleve = 'oui' WHERE
((table_choix_ic.tige_preleve)=' & no_tige & ');"
        CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui'
WHERE ((tige_a_couper.id_TIGE)=' & no_tige & ');"
    End If

' recommencer la liste des arbre martele si ont a visi ter tout les arbres

```

```

If x = UBound(liste_tige_a_preleve, 2) Then
    Set rs_tige_a_preleve = New ADODB.Recordset
    rs_tige_a_preleve.Open "SELECT table_choix_ic.tige_martele AS ID_M,
Max(table_choix_ic.IC) AS IC, First(table_choix_ic.dhp_tige_prelev) AS DHP_P,
First(table_choix_ic.tige_preleve) AS ID_P, First(table_choix_ic.st_tige_ha) AS st_P
FROM table_choix_ic WHERE (((table_choix_ic.preleve)='non')) GROUP BY
table_choix_ic.tige_martele ORDER BY Rnd(Now()*table_choix_ic.tige_martele);", CONN,
adOpenStatic, adLockOptimistic
    If rs_tige_a_preleve.EOF Then
        rs_tige_a_preleve.Close
        Set rs_tige_a_preleve = Nothing
        Exit For
    End If
    liste_tige_a_preleve = rs_tige_a_preleve.GetRows
    rs_tige_a_preleve.Close
    Set rs_tige_a_preleve = Nothing
    x = -1
End If

Next
a = 1
's'il n'y a plus de tige dans un rayon de 10m et que le but n'est pas atteint alors prendres
les tiges au hasard
Set somme_st_preleve = New ADODB.Recordset
somme_st_preleve.Open "SELECT Sum([tige_a_couper]![st_tige_ha_orig]) AS st_ha
FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui') AND
((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
st_preleve_tot = somme_st_preleve(0)
somme_st_preleve.Close
Set somme_st_preleve = Nothing
Set rs_st_orig = New ADODB.Recordset
rs_st_orig.Open "SELECT Sum(tbl.st_tige_ha_orig) AS Expr1 FROM tige_a_couper as tbl
WHERE (((tbl.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
but_st = (rs_st_orig(0) * taux_prelev)
rs_st_orig.Close
Set rs_st_orig = Nothing
If st_preleve_tot < but_st Then
    Set rs_tige_a_preleve = New ADODB.Recordset

```

```

rs_tige_a_preleve.Open "SELECT tige_a_couper.id_tige,
tige_a_couper.st_tige_ha_orig FROM tige_a_couper WHERE
(((tige_a_couper.tige_preleve)='non') AND ((tige_a_couper.martele)='non') AND
((tige_a_couper.buffer)='orig')) ORDER BY Rnd(Now()*tige_a_couper.id_tige);", CONN,
adOpenStatic, adLockOptimistic
liste_tige_a_preleve = rs_tige_a_preleve.GetRows
rs_tige_a_preleve.Close
Set rs_tige_a_preleve = Nothing
For u = 0 To UBound(liste_tige_a_preleve, 2)
    CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui'
WHERE (((tige_a_couper.id_TIGE)=" & liste_tige_a_preleve(0, u) & "));"
    Set somme_st_preleve = New ADODB.Recordset
    somme_st_preleve.Open "SELECT Sum([tige_a_couper].[st_tige_ha_orig]) AS
st_ha FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui') AND
((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
    st_preleve_tot = somme_st_preleve(0)
    somme_st_preleve.Close
    Set somme_st_preleve = Nothing
    If st_preleve_tot > but_st Then
        Exit For
    End If
Next
End If
Set somme_st_preleve = New ADODB.Recordset
somme_st_preleve.Open "SELECT Sum([tige_a_couper].[st_tige_ha_orig]) AS st_ha
FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui') AND
((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
st_preleve_tot = somme_st_preleve(0)
somme_st_preleve.Close
Set somme_st_preleve = Nothing
If st_preleve_tot > but_st + 0.5 Then
    'trier les tige deja preleve qui font le moins de ic
    Set rs_tige_a_preleve = New ADODB.Recordset
    rs_tige_a_preleve.Open "SELECT tige_a_couper.id_tige, table_choix_ic.IC,
tige_a_couper.tige_preleve, tige_a_couper.martele, tige_a_couper.buffer FROM
tige_a_couper LEFT JOIN table_choix_ic ON tige_a_couper.id_tige =
table_choix_ic.tige_preleve WHERE (((tige_a_couper.tige_preleve)='oui') AND

```

```

((tige_a_couper.martele)='non') AND ((tige_a_couper.buffer)='orig')) ORDER BY
table_choix_ic.IC;", CONN, adOpenStatic, adLockOptimistic
    liste_tige_a_preleve = rs_tige_a_preleve.GetRows
    rs_tige_a_preleve.Close
    Set rs_tige_a_preleve = Nothing
    For x = 0 To UBound(liste_tige_a_preleve, 2)
        CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'non'
WHERE ((tige_a_couper.id_TIGE)=" & liste_tige_a_preleve(0, x) & ") ;"
        Set somme_st_preleve = New ADODB.Recordset
        somme_st_preleve.Open "SELECT Sum([tige_a_couper]![st_tige_ha_orig]) AS
st_ha FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui') AND
((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
        st_preleve_tot = somme_st_preleve(0)
        somme_st_preleve.Close
        Set somme_st_preleve = Nothing
        If st_preleve_tot < but_st + 0.5 Then
            Exit For
        End If
    Next
End If
Set somme_st_preleve = New ADODB.Recordset
somme_st_preleve.Open "SELECT Sum([tige_a_couper]![st_tige_ha_orig]) AS st_ha
FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui') AND
((tige_a_couper.buffer)='orig'));", CONN, adOpenStatic, adLockOptimistic
st_preleve_tot = somme_st_preleve(0)
somme_st_preleve.Close
Set somme_st_preleve = Nothing
'verification si je ne retombe pas en dessous de ma limite et verification pour ne pas
tomber dans une boucle sans fin
If st_preleve_tot < but_st And boucle < 20 Then
    'on recommence
    boucle = boucle + 1

    GoTo recommence
End If

Set rs_st_orig = Nothing

```



```

table_IC_avant.no_tige_sujet;"

End Sub
Public Sub calcul_IC_apres(martelage, prelev, parcelle, no_simul)
On Error Resume Next
Set rs_liste_tige = New ADODB.Recordset
rs_liste_tige.Open "SELECT tige_a_couper.NO_TIGE, tige_a_couper.DHP,
tige_a_couper.X, tige_a_couper.Y, tige_a_couper.buffer, tige_a_couper.martele,
tige_a_couper.tige_preleve FROM tige_a_couper;", CONN, adOpenStatic,
adLockOptimistic
liste_tige = rs_liste_tige.GetRows
rs_liste_tige.Close
Set rs_liste_tige = Nothing
'tige_martele(0,..) = ID de la tige
'tige_martele(1,..) = DHP de la tige
'tige_martele(2,..) = position X de la tige
'tige_martele(3,..) = position Y de la tige
'tige_martele(4,..) = buffer de la tige
'tige_martele(5,..) = martele oui/non
'tige_martele(6,..) = preleve oui/non
conn_mysql.Execute "DROP TABLE IF EXISTS table_IC_apres;"
conn_mysql.Execute "CREATE TABLE table_IC_apres (no_tige_sujet
char(20),no_tige_comp char(20),compet DOUBLE,martele char(5), prelev char(5));"
superficie_total = val_pi * 100
For x = 0 To UBound(liste_tige, 2)
  If liste_tige(4, x) = "orig" Then
    conn_mysql.Execute "INSERT INTO table_IC_apres ( no_tige_sujet ,no_tige_comp
,compet,martele,prelev ) SELECT '" & liste_tige(0, x) & "' , ",0,'" & liste_tige(5, x) & "','"
& liste_tige(6, x) & "' ;"
    If liste_tige(6, x) = "non" Then
      For y = 0 To UBound(liste_tige, 2)
        If ((liste_tige(2, x) - liste_tige(2, y)) ^ 2 + (liste_tige(3, x) - liste_tige(3,
y)) ^ 2) ^ (1 / 2) < 10 And liste_tige(0, x) <> liste_tige(0, y) And liste_tige(1, x) * 0.75
<= liste_tige(1, y) And liste_tige(6, y) = "non" Then
          st_tige = (val_pi * (liste_tige(1, y) / 2 / 100) ^ 2) * 10000 / (val_pi *
(rs_rayon(0) ^ 2))

```

```

        conn_mysql.Execute "INSERT INTO table_IC_apres ( no_tige_sujet
, no_tige_comp ,compet,martele,prelev ) SELECT "" & liste_tige(0, x) & "" , "" &
liste_tige(0, y) & "" , "" & st_tige & "" , "" & liste_tige(5, x) & "" , "" & liste_tige(6, x) & "" ;"
        End If
    Next
End If
End If
Next
conn_mysql.Execute "INSERT INTO compilation_IC_apres ( no_tige, parcelle, no_simul,
martelage, prelevement, type_tige, compet ) SELECT table_IC_apres.no_tige_sujet, "" &
parcelle & "" , "" & no_simul & "" , "" & martelage & "" , "" & prelev & "" ,
If(table_IC_apres.martele='oui','martele+',If(table_IC_apres.prelev='oui','martele -
','bourage')), Sum(table_IC_apres.compet) FROM table_IC_apres GROUP BY
table_IC_apres.no_tige_sujet;"
End Sub
Public Sub enregistrer_resultats(tx_martelage, tx_preleve, parcelle, no_simul)
On Error Resume Next
superficie_total = (arete * 4) ^ 2
Set rs_st_init_tout = New ADODB.Recordset
rs_st_init_tout.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2)) AS
st FROM tige_a_couper;", CONN, adOpenStatic, adLockOptimistic
initial_tout = rs_st_init_tout(0) * 10000 / superficie_total
rs_st_init_tout.Close
Set rs_st_init_tout = Nothing
Set rs_st_martele_tout = New ADODB.Recordset
rs_st_martele_tout.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2))
AS st FROM tige_a_couper WHERE (((tige_a_couper.martele)='oui'));", CONN,
adOpenStatic, adLockOptimistic
st_martele_tout = rs_st_martele_tout(0) * 10000 / superficie_total
rs_st_martele_tout.Close
Set rs_st_martele_tout = Nothing
Set rs_st_preleve_tout = New ADODB.Recordset
rs_st_preleve_tout.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2))
AS st FROM tige_a_couper WHERE (((tige_a_couper.tige_preleve)='oui'));", CONN,
adOpenStatic, adLockOptimistic
st_preleve_tout = rs_st_preleve_tout(0) * 10000 / superficie_total
rs_st_preleve_tout.Close
Set rs_st_preleve_tout = Nothing

```

```

Set rs_st_init_parcelle = New ADODB.Recordset
rs_st_init_parcelle.Open "SELECT Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2))
AS st FROM tige_a_couper WHERE (((tige_a_couper.buffer)='orig'))";, CONN,
adOpenStatic, adLockOptimistic
initial_parcelle = rs_st_init_parcelle(0) * mult_ha
rs_st_init_parcelle.Close
Set rs_st_init_parcelle = Nothing
Set rs_st_martele_parcelle = New ADODB.Recordset
rs_st_martele_parcelle.Open "SELECT
Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2)) AS st FROM tige_a_couper WHERE
(((tige_a_couper.martele)='oui') AND ((tige_a_couper.buffer)='orig'))";, CONN,
adOpenStatic, adLockOptimistic
st_martele_parcelle = rs_st_martele_parcelle(0) * mult_ha
rs_st_martele_parcelle.Close
Set rs_st_martele_parcelle = Nothing
Set rs_st_preleve_parcelle = New ADODB.Recordset
rs_st_preleve_parcelle.Open "SELECT
Sum(((Atn(1)*4)*([tige_a_couper]![DHP]/2/100)^2)) AS st FROM tige_a_couper WHERE
(((tige_a_couper.tige_preleve)='oui') AND ((tige_a_couper.buffer)='orig'))";, CONN,
adOpenStatic, adLockOptimistic
st_preleve_parcelle = rs_st_preleve_parcelle(0) * mult_ha
rs_st_preleve_parcelle.Close
Set rs_st_preleve_parcelle = Nothing
conn_mysql.Execute "INSERT INTO compilation ( parcelle, taux_prelevement,
taux_martelage, no_simul, ST_initial_tout, ST_martele_tout, ST_preleve_tout,
ST_initial_parcelle, ST_martele_parcelle, ST_preleve_parcelle ) SELECT "" & parcelle & ""
, "" & tx_preleve & "", "" & tx_martelage & "", "" & no_simul & "", "" & initial_tout & "", "" &
st_martele_tout & "", "" & st_preleve_tout & "", "" & initial_parcelle & "", "" &
st_martele_parcelle & "", "" & st_preleve_parcelle & "" ;"
End Sub
Private Sub atteinte_but_martelage(no_tuile, nb_tuile_noncoupe, pourc_but, tbl_tige1)
verif1 = no_tuile & ";" & nb_tuile_noncoupe & ";" & pourc_but
ordre_prelev = 1
Set rs_but_st = New ADODB.Recordset
rs_but_st.Open "SELECT st_cellule.but FROM st_cellule WHERE
(((st_cellule.numero_cellule)=" & no_tuile & ")");, CONN, adOpenStatic,
adLockOptimistic
but_st = rs_but_st(0)

```

```

rs_but_st.Close
Set rs_but_st = Nothing
' condition si le but de recolte de cette tige est negatif, alors il ne faut pas coupé et sortir
immédiatement
If but_st < 0 Then
    CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'oui' WHERE
((st_cellule.numero_cellule)=" & no_tuille & ") ;"
    Exit Sub
End If
verif1 = verif1 & ";" & but_st & ";" & UBound(tbl_tige1, 2) + 1
'verification si une tuile contient seulement une tige ou zero tige
If UBound(tbl_tige1, 2) = 0 Then
    st_enleve = tbl_tige1(8, 0)
    CONN.Execute "UPDATE tige_a_couper SET tige_a_couper.tige_preleve = 'oui' WHERE
(((tige_a_couper.NO_TIGE)=" & tbl_tige1(1, tige_extraite) & ") AND
((tige_a_couper.buffer)="" & tbl_tige1(6, tige_extraite) & ""));"
    verif = verif1 & ";" & (tbl_tige1(1, 0)) & ";" & (tbl_tige1(6, 0)) & ";" & (tbl_tige1(8,
0)) & ";" ;"
    verif = ""
ElseIf IsEmpty(tbl_tige1) Then
    st_enleve = 0
    verif = verif1 & ";" & ";" & ";" & ";" & ";" ;"
    verif = ""
Else

st_precedent = 0
For t = 0 To UBound(tbl_tige1, 2)

    nb_tiges = UBound(tbl_tige1, 2) + 1
    st_enleve = tbl_tige1(7, t) + st_precedent
    If st_enleve < but_st Then
        CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele =
'oui' WHERE (((cellules_martelage.NO_TIGE)=" & tbl_tige1(1, t) & ") AND
((cellules_martelage.buffer)="" & tbl_tige1(6, t) & ""));"
        st_precedent = st_enleve

```

```

    verif = verif1 & ";" & (tbl_tige1(1, t)) & ";" & (tbl_tige1(6, t)) & ";" &
(tbl_tige1(8, t)) & ";recolte;";
    verif = ""
    'apres avoir martele une tige et que lon veut en martele une autre aller voir la
fonction qui permet de choisir les classes de dhp
    tbl_tige1 = condition_classe(no_tuille)
    '
    Call condition_classe(no_tuille)
    t = -1
    If IsNull(tbl_tige1) Or IsEmpty(tbl_tige1) Then
        Exit For
    End If
    ElseIf st_enleve >= but_st And (Abs(but_st - st_precedent) >= Abs(st_enleve -
but_st)) Then
        CONN.Execute "UPDATE cellules_martelage SET cellules_martelage.martele =
'oui' WHERE (((cellules_martelage.NO_TIGE)=" & tbl_tige1(1, t) & ") AND
((cellules_martelage.buffer)=" & tbl_tige1(6, t) & "));";
        st_precedent = st_enleve
        verif = verif1 & ";" & (tbl_tige1(1, t)) & ";" & (tbl_tige1(6, t)) & ";" &
(tbl_tige1(8, t)) & ";recolte au-dessus;";
        verif = ""
        Exit For
    ElseIf st_enleve >= but_st And (Abs(but_st - st_precedent) < Abs(st_enleve -
but_st)) Then
        st_enleve = st_precedent
        verif = verif1 & ";" & (tbl_tige1(1, t)) & ";" & (tbl_tige1(6, t)) & ";" &
(tbl_tige1(8, t)) & ";non-recolte;";
        verif = ""
        Exit For
    End If
Next
End If
'a la fin de la visite de la tuile je repartie ce qui a ete coupe en trop ou ce qui manque à
coupé
Set rs_st_recolte = New ADODB.Recordset
rs_st_recolte.Open "SELECT Sum(cellules_martelage.st_tige_ha) AS SumOfst_tige FROM
cellules_martelage WHERE (((cellules_martelage.numero_cellule)=" & no_tuille & ") AND
((cellules_martelage.martele)='oui'))";, CONN, adOpenStatic, adLockOptimistic
If IsNull(rs_st_recolte(0)) Then st_recolte = 0 Else st_recolte = rs_st_recolte(0)

```

```

rs_st_recolte.Close
Set rs_st_recolte = Nothing
st_a_repartir = but_st - st_recolte
If nb_tuile_noncoupe = 0 Then
    st_reparti_final = st_a_repartir
Else
    st_a_repartir_voisin = st_a_repartir / nb_tuile_noncoupe
    CONN.Execute "UPDATE st_cellule SET st_cellule.coupe_oui_non = 'oui' WHERE
((st_cellule.numero_cellule)=" & no_tuile & ") ;"
    CONN.Execute "UPDATE st_cellule SET st_cellule.but = [st_cellule]![but]+(" &
st_a_repartir_voisin & ") WHERE (((st_cellule.coupe_oui_non)='non'));"
End If
verif = verif1 & ";;;;;;;;;" & st_a_repartir & ";" & st_a_repartir_voisin
verif = ""
End Sub

```

Modules vb6

```

Type Tableau
    ligne() As String
    col() As String
End Type
Public rs_rayon As ADODB.Recordset
Public CONN As ADODB.Connection
Public conn_mysql As ADODB.Connection
Public rs As ADODB.Recordset
Public parcelle_data() As String
Public nb_alea() As String
Public parcelle_all()
Public ajout
Public taille_array
Public carre_a
Public carre_b
Public carre_c
Public carre_d
Public arete
Public FN_database
Public val_pi
Public numero_simul
Public liste_table

```

```
Public nomtable_output
Public niveau_prelev
Public parcelle_recolte
Public mult_ha
Public compt_prelev
Public lst_prelev
Public lst_parcelle
Public array_tuille
Public parcelle_active
Public nb_tuille_tot
Public pourc_but
Public tbl_tige1
Public liste_cond()
Public Sub ouvrir_connection()
FN_database = Form1.Text1.Text
Set CONN = New ADODB.Connection
Provider = "Microsoft.JET.OLEDB.4.0"
Connexion = FN_database
CONN.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source= " &
Connexion
CONN.Open
End Sub
Public Sub ouvrir_connection_mysql()
Set conn_mysql = New ADODB.Connection
conn_mysql.CursorLocation = adUseClient
conn_mysql.ConnectionString = "DRIVER={MySQL ODBC 3.51 Driver};" _
& "SERVER=localhost;" _
& "DATABASE=ccfb_nb;" _
& "UID=root;" _
& "PWD=vi0l0n;" _
& "OPTION=" & 1 + 2 + 8 + 32 + 2048 + 16384

conn_mysql.Open
End Sub
Public Sub delete_table_if_exist(table As String)
Set rs = CONN.OpenSchema(adSchemaTables)
While Not rs.EOF
If rs.Fields("TABLE_TYPE") = "TABLE" And rs!TABLE_NAME = table Then
```

```
        CONN.Execute "drop table " & table
    End If
    rs.MoveNext
Wend
rs.Close
End Sub
Public Sub delete_table_if_exist_mysql(table As String)
Set rs = conn_mysql.OpenSchema(adSchemaTables)
While Not rs.EOF
    If rs.Fields("TABLE_TYPE") = "TABLE" And rs!TABLE_NAME = table Then
        conn_mysql.Execute "drop table " & table
    End If
    rs.MoveNext
Wend
rs.Close
End Sub
Public Sub ouvrir_table(condition)
nom_table = Form1.tbl_choisi_form.Text
Call delete_table_if_exist("parcelle_data")
requete = "SELECT " & nom_table & ".Parcelle, " & nom_table & ".NO_TIGE, " &
nom_table & ".ESSENCE, " & nom_table & ".DHP, " & nom_table & ".X, " & nom_table &
".Y INTO parcelle_data FROM " & nom_table & " WHERE (((" & nom_table & ".Parcelle)="
& condition & "));"
CONN.Execute requete
End Sub
Public Sub nb_aleatoire(min As Integer, max As Integer, nombre As Integer)
Randomize Timer
ReDim nb_alea(nombre - 1)
i = 0
While i < nombre
    presence = False
    valeur_aleat = Int(Rnd() * (max - min + 1)) + min
    For j = 0 To UBound(nb_alea)
        If nb_alea(j) = "" Then
            ElseIf nb_alea(j) = valeur_aleat Then
                presence = True
        End If
    Next
    i = i + 1
End While
End Sub
```

```

    If presence = False Then
        nb_alea(i) = valeur_aleat
        i = i + 1
    End If
Wend
End Sub
Public Function aleatoire(min, max)
    Randomize Timer
    aleatoire = Int(Rnd() * (max - min + 1)) + min
End Function
Public Function condition_classe(no_tuille)
'section pour choisir la classe de dhp
    Set rs_classe_deja_martele = New ADODB.Recordset
    rs_classe_deja_martele.Open "SELECT cellules_martelage.classe_dhp From
cellules_martelage Where (((cellules_martelage.DHP) < 40) And
((cellules_martelage.numero_cellule) = " & no_tuille & ")And
((cellules_martelage.martele) = 'oui'))";", CONN, adOpenStatic, adLockOptimistic
    'creation dune condition permettant de choisir une nouvelle classe de dhp
    If Not rs_classe_deja_martele.EOF Then
        rs_classe_deja_martele.MoveFirst
        rs_classe_deja_martele.MoveLast
        rs_classe_deja_martele.MoveFirst
    End If
    cond1 = ""
    Do While rs_classe_deja_martele.EOF = False
        cond1 = cond1 & "And ((cellules_martelage.martele) <> "" &
rs_classe_deja_martele(0) & "")"
        rs_classe_deja_martele.MoveNext
    Loop
    rs_classe_deja_martele.Close
    Set rs_tige_cellule = New ADODB.Recordset
    rs_tige_cellule.Open "SELECT cellules_martelage.* FROM cellules_martelage
WHERE (((cellules_martelage.numero_cellule)=" & no_tuille & ")) AND
((cellules_martelage.martele)='non')" & cond1 & " ORDER BY
Rnd(Now()*cellules_martelage.no_tige),Rnd(Now()*cellules_martelage.x),Rnd(Now()*cel
lules_martelage.y);", CONN, adOpenStatic, adLockOptimistic
    'verification si il y des tiges dans les classe d'age restante sinon on reprend tout les
classe

```

```

    If rs_tige_cellule.EOF Then
        rs_tige_cellule.Close
        rs_tige_cellule.Open "SELECT cellules_martelage.* FROM cellules_martelage
WHERE (((cellules_martelage.numero_cellule)=" & no_tuile & ")) AND
((cellules_martelage.martele)='non') ORDER BY
Rnd(Now()*cellules_martelage.no_tige),Rnd(Now()*cellules_martelage.x),Rnd(Now()*cel
lules_martelage.y);", CONN, adOpenStatic, adLockOptimistic
        If Not rs_tige_cellule.EOF Then
            tbl_tige1 = rs_tige_cellule.GetRows
        Else
            tbl_tige1 = Null
        End If
    Else
        tbl_tige1 = rs_tige_cellule.GetRows
    End If
    rs_tige_cellule.Close
    condition_classe = tbl_tige1
End Function
Public Sub condition_classe_all(liste_condi, classe_dhp)
If Not classe_dhp = "" Then
    If liste_condi(0) = "" Then
        liste_condi(0) = classe_dhp
    ElseIf liste_condi(1) = "" Then
        liste_condi(1) = classe_dhp
    Else
        liste_condi(0) = ""
        liste_condi(1) = ""
    End If
End If
End Sub
Public Sub condition_classe_demartele(classe_dhp)
End Sub
Public Function CompactDB(pFileName As String) As Boolean
'-----
' PROCEDURE : CompactDB
' DESCRIPTION :
' RETURN VALUE: true or false depending wether there were errors or not
' NOTE :

```

' Add "Microsoft Jet and Replication Objects 2.x Library" to
' the references of your project

' EXAMPLE :

```
'-----  
On Error GoTo ErrH  
Dim CONN As New JRO.JetEngine  
Dim ConnstringSorg As String, ConnstringDest As String  
' Ensure file is not read only  
SetAttr pFileName, vbNormal  
ConnstringSorg = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _  
pFileName & ";User ID=;Password=;"  
ConnstringDest = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _  
App.Path & "\Temp.mdb" & ";Jet OLEDB:Engine Type=5;"  
Screen.MousePointer = vbHourglass  
CONN.CompactDatabase ConnstringSorg, ConnstringDest  
Screen.MousePointer = vbDefault  
'Copia il file compattato.  
Kill pFileName  
FileCopy App.Path & "\Temp.mdb", pFileName  
Kill App.Path & "\Temp.mdb"  
Set CONN = Nothing  
CompactDB = True  
Exit Function  
ErrH:  
Screen.MousePointer = vbDefault  
'debug.print Err.Description  
End Function
```