Q : Quelle est la pertinence de la structure de données choisie dans ce laboratoire ? Est-ce que cette représentation des données est supérieure à un autre choix de TDA ? Quels sont les avantages et les inconvénients du TDA choisi par rapport à un autre TDA (de votre choix) ?R : L’arbre naire a pour principal avantage to pouvoir d’avoir plusieurs éléments rattachés à un même nœud. Ceci nous est particulièrement utile pour faire le lien entre une branche parent et ses nombreux enfants. En effet, nous ne savons pas combiens d’enfant une branche aura avant l’exécution du programme, il est donc important que le nombre d’enfant que peut accepter un nœud soit variable, comme c’est le cas pour la TDA Arbre. Dans mon implémentation de l’arbre pour ce laboratoire, les nœuds ont également un chemin d’accès vers leur parent afin de pouvoir obtenir de l’information sur leur position dans l’espace. Ceci évite de devoir conserver ses informations à même le nœud est devoir faire du « forward propagation » lorsque cette information a changé.

Q : Comment pourrait-on modifier la structure de données que vous avez définie afin de permettre l’addition de feuilles, de fruits ou d’autres éléments qui apparaissent aléatoirement aux dernières branches de l’arbre ?

R : Puisque les nœuds utilisent des pointeurs pour se référencer entre eu, nous pourrions ici directement utiliser la structure déjà en place et des éléments aléatoires des classes descendantes de Branch et écraser la fonction d’affichage pour afficher la bonne forme et la fonction de simulation si nous voulons que ses éléments se comportent différemment. Nous aurions ainsi les deux requis pour utiliser le polymorphisme, soit un lien d’héritage entre les classes et un appel au travers d’une référence, ici un pointeur. Donc non, la structure n’a pas besoins de changer elle-même, c’est la façon de créer les autres éléments qui doit être adaptée.

Q : Chez un arbre réel, il est possible que des branches seules ou des groupes de branches apparaissent après une longueur constante sur un tronc, ou simplement à des longueurs plus ou moins aléatoires. Nous voulons permettre que des branches sortent d’un nœud qui se trouverait au milieu (et non pas à l’extrémité) d’une branche. Expliquez comment cela est réalisable par une modification de votre structure de données.

R : La structure elle-même peut être conservée, il suffirait d’implémenter une fonction qui sectionnerait une branche à une longueur aléatoire, créant ainsi deux branches liées ensemble en un nœud. De nouvelles branches plus petites peuvent alors être ajoutée à ce nouveau nœud. Puisque la résistance en flexion d’une branche dépend en grande partie de son diamètre et que celui-ci serait constant des deux côtés du nouveau nœud, la dynamique du système ne devrait pas trop être affectée.