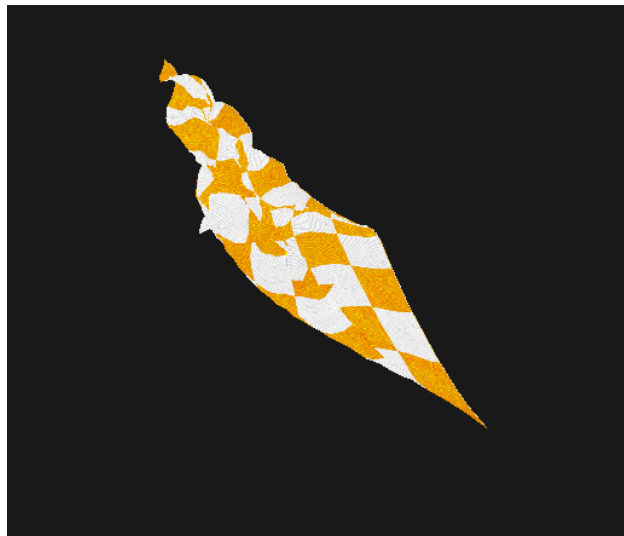


Rapport du TP2 du module Animation et Simulation

Animation de tissus

Antoine BRALET et Guillaume DURET



3 Décembre 2020

1 Introduction

Ce rapport vise à rendre compte des différents éléments implémentés mais aussi de détailler les problèmes rencontrés, les solutions trouvées ainsi que les explications algorithmiques et physiques de ceux-ci dans le cadre du TP d'animation de tissus du module d'Animation et Simulation.

Nous allons en particulier nous intéresser ici dans un premier temps, en section 2, au comportement d'un ressort et aux forces auxquelles il est soumis lorsqu'une masse est appliquée à l'une de ses extrémités quand l'autre reste fixée. L'idée ici est ensuite de pouvoir simuler un tissu à partir d'une grille déformable dont chaque point est relié à ses voisins par un ressort. Le détail exacte de cette simulation se fait en section 3. Puis de nouvelles contraintes sous la forme de collisions avec un plan et une sphère ou encore la présence de vent sont ajoutées dans la section 4, avant de modéliser l'ensemble du phénomène sur GPU à partir d'un *Compute Shader*.

Notons ici qu'afin d'exécuter le code, il est nécessaire de reprendre l'ensemble des fichiers disponible dès le début du TP et de remplacer seulement certains dossiers par les dossiers qui fournis dans l'archive jointe à ce travail. Leurs noms sont restés volontairement identique afin qu'il soit plus aisé de trouver et remplacer ces fichiers dans l'archive de départ du TP.

2 Simulation d'un ressort en 3D

2.1 Rappel du problème

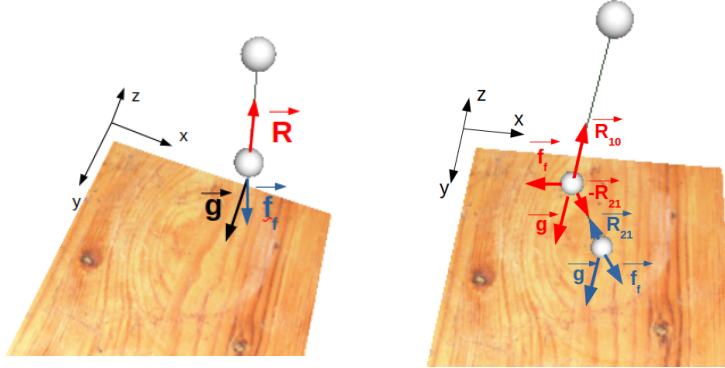


Figure 1: Schéma permettant de visualiser le bilan des forces sur un et deux ressorts respectivement à gauche et à droite, en 3 dimensions (en considérant que le point du haut est fixé à un plafond invisible et que le sol se situe au niveau du plancher). \vec{R} représente la force de rappel, \vec{g} la gravité et \vec{f}_f la force de frottements fluides appliquée en l'extrémité courante. Sur l'image de droite, les indices utilisés avec les forces de rappel représentent la direction de la force : \vec{R}_{ij} représente la force provoquée par l'extrémité j sur i .

Notons tout d'abord que le cas d'un ressort 3D s'extrapole très bien à partir du cas d'un ressort 2D. L'équation 1 de la force de rappel en 2D nous permet ainsi de facilement déterminer l'équation 2 décrivant la force de rappel d'un ressort en 3D en considérant R la force de rappel, K la constante de raideur du ressort, L_0 la longueur à vide du ressort, $p(t)$ la position de la masse à l'instant t dans l'espace 2D et $u(t)$ et $\|u(t)\|$ respectivement le vecteur reliant les deux extrémités du ressort au temps t et sa norme.

$$R = K.(p(t) - L_0) \quad (1)$$

$$R = K.(||u(t)|| - L_0) \cdot \frac{u(t)}{||u(t)||} \quad (2)$$

De cette façon il est alors possible de déterminer le comportement de la masse du ressort en faisant un bilan des forces présentes sur l'image de gauche de la figure 1. À savoir :

- La force de rappel décrite par l'équation 2
- La force de la gravité : $\vec{g} = -m.g.\vec{z}$
- La force de frottements fluides : $\vec{f}_f = -\mu.\vec{v}(t)$

avec m la masse de la masse de l'extrémité du ressort, g la constante de gravité : 9.81, μ le coefficient d'amortissement et $\vec{v}(t)$ le vecteur vitesse de la masse au temps t .

Ce faisant il est alors possible de déterminer les équations du mouvement à partir des lois de Newton, en continu puis discrétisées suivant l'équation 3 que nous cherchons à implémenter et paramétrer dans la section 2.2.

$$\begin{cases} m \cdot \frac{dv}{dt} = R(t) + f_f(t) + g & \rightarrow v_{t+1} = v_t + \frac{\Delta t}{m} \cdot (R(t) + f_f(t) + g) \\ \frac{dp}{dt} = v(t) & \rightarrow p_{t+1} = p_t + \Delta t \cdot v_t \end{cases} \quad (3)$$

2.2 Simulation numérique

L'implémentation de la simulation numérique d'un ressort avait déjà été faite lors du début du TP. Après vérification du bon fonctionnement de celle-ci ayant permis l'obtention de l'image de gauche de la Figure 1. Nous allons alors ici chercher à modifier les paramètres afin de vérifier le comportement du ressort et/ou de la stabilité et la réalité de la simulation réalisée. Trois paramètres en particuliers sont étudiés ici : la constante de raideur du ressort K , le coefficient d'amortissement μ et le pas de temps d'intégration Δt .

Modification de la constante de raideur K La modification de cette constante de raideur peut faire apparaître plusieurs comportements. La simulation stable présentée sur l'image de gauche de la Figure 1 a été réalisée pour $K = 20$. Ainsi, en considérant $K = 5$, on peut noter que le ressort tend à s'allonger très fortement et nécessite une distance entre la masse et l'extrémité fixe très grande avant de voir la force de rappel agir sur le ressort. Ce résultat est assez cohérent avec la réalité, excepté le fait que à partir d'une certaine longueur le ressort devrait ne plus pouvoir s'étirer davantage. Or si K devient vraiment petit alors le ressort peut s'étirer sur une très grande distance ce qui est impossible en réalité de par les caractéristiques physiques du ressort. Il est également possible de constater un phénomène complètement différent dans le cas où le ressort a une constante de raideur très forte. En effet en plaçant $K = 1000$, il est possible de noter que notre ressort tend à osciller autour de la position stable. Le problème n'est pas tant le fait qu'il oscille mais le fait que l'amortissement de ces oscillations soit très long, ce qui n'est pas cohérent avec la réalité. En effet, lorsque la constante de raideur est très forte, en réalité, le ressort aurait tendance à vite revenir à sa position stable avec une ou deux oscillations au maximum. Plus improbable encore, il est possible de modifier la position d'origine de la masse du ressort. En effet, jusqu'à présent, cette position était (0.0, 0.0, 0.4) ce qui était très proche de la longueur du ressort à vide. Mais si l'on modifie cette position à (0.0, 0.0, 1.0), en conservant la constante de raideur très forte, alors on peut noter que notre ressort oscille dès le départ et ne convergera jamais car l'amortissement ne se fait pas, ce qui rend le mouvement complètement irréaliste. Attention cependant à ne pas faire d'amalgames, la très forte constante de raideur n'est pas la seule raison pour laquelle le résultat diverge, le pas de temps d'intégration l'est également, mais nous verrons cela dans le dernier paragraphe de cette section.

Modification du coefficient d'amortissement μ Ce coefficient d'amortissement permet de faire en sorte que le ressort ait un mouvement particulièrement réaliste, à savoir, qu'il finisse par converger. En effet si l'on place cette constante à 0, il est notable que le résultat ne convergera pas. Ceci pourrait donc être considéré comme vrai si le ressort était placé dans le vide mais dans toutes autres circonstances, ce résultat serait complètement faux. Ainsi, le coefficient d'amortissement est particulièrement dépendant du matériau dans lequel se déplace le ressort. Ceci implique que ce coefficient sera bien supérieur dans l'eau que dans l'air. Il dépend en réalité énormément de la densité du matériau traversé. Il est alors notable que ce phénomène est également reproduit lors de la simulation numérique. Ceci en prenant garde tout de même au fait qu'il ne faille pas qu'il soit trop petit pour éviter de ne jamais converger. Mais attention également à un μ trop grand. En effet lorsque l'on choisit $\mu = 100$ contrairement au paramètre initial de 0.5, on peut alors noter de nouveau un phénomène d'oscillations qui apparait dans la simulation alors que le phénomène devrait tout au plus se rapprocher très rapidement de sa position de stabilité, nous soupçonnons de nouveau que la cause de ceci soit en grande partie liée au paramètre d'intégration temporelle décrit dans le paragraphe suivant. De plus il est remarquable qu'un autre phénomène n'est pas pris en compte ici : si la densité est très élevée comme c'est le cas ici, dans ce cas le ressort ne devrait pas pouvoir se mouvoir dans le milieu car le milieu est trop dense. Un exemple serait de mettre un ressort dans du ciment et attendre qu'il réagisse. Dans ce cas la densité est tellement forte que le ressort devrait garder sa position initiale, ce qui n'est pas le cas dans cette simulation, très certainement parce que la force d'Archimède notamment n'est pas prise en compte dans la simulation.

Modification du pas de temps Δt Le pas d'intégration temporelle est un paramètre très important pour la simulation car il détermine à quel pas de temps les positions, vitesses et forces vont être actualisées. Dans la réalité cette actualisation se fait en continu, ici pour des raisons numériques, nous sommes forcés de discrétiser. Et en particulier un pas de temps trop faible aura tendance à ralentir beaucoup la simulation tout en restant tout de même cohérent. Néanmoins, aussi infime soit-il nous faisons tout de même une approximation, l'effet de cette approximation est d'autant plus visible lorsque le pas de temps est trop grand. Deux situations de pas "trop grand" sont observables : premièrement si le pas est de $\Delta t = 0.2$ contrairement à 0.02 dans la simulation initiale, on peut voir le ressort converger très vite vers sa position finale mais sans passer par les états intermédiaires caractéristiques du mouvement du ressort. Deuxièmement, le cas où $\Delta t = 0.45$ où l'on peut remarquer très rapidement que la simulation diverge. L'explication de ce phénomène provient du fait que nous utilisons la méthode d'Euler explicite afin de discrétiser notre problème dans l'équation 3. Alors le problème qui survient est le même que pour le cas d'école de la chute libre illustré en Figure 2.

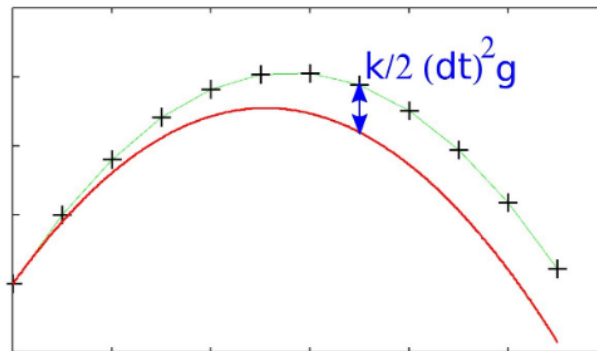


Figure 2: Illustration des problématiques d'approximation liées à la méthode de discrétisation d'Euler explicite. Cas sur la chute libre vu en cours d'Animation et Simulation et applicable à notre simulation de ressorts (en adaptant les forces en présence). Ici en rouge, la trajectoire réelle de la masse en chute libre, en vert la trajectoire tracée par la méthode d'Euler explicite.

Plusieurs possibilités s'offrent à nous pour régler ce type de problèmes. Le premier d'entre eux consiste évidemment à choisir un pas d'intégration suffisamment faible pour que la simulation reste acceptable (ce que nous ferons par ailleurs pour la suite de ce TP en considérant $\Delta t = 0.02$). Il est également possible de se servir de la méthode d'Euler implicite qui "sous-estime" la trajectoire. Ainsi comme la méthode d'Euler explicite la "sur-estime", il pourrait être envisageable d'interpoler les deux résultats obtenus afin d'avoir une simulation plus proche de la réalité. Il nous serait également possible de mettre en place une autre méthode dite de Runge-Kutta d'ordre k qui consiste à séparer le pas de temps en k sous-pas de temps afin de limiter les dérives de la discrétisation.

2.3 Ressorts couplés

En guise d'approche pour la section 3 qui suit et qui aborde directement la simulation de tissus, il est possible d'observer le comportement de deux ressorts mis bout à bout en fonction des forces qui s'y applique. Comme précédemment, il est possible de trouver le schéma de la simulation avec les forces en présence sur l'image de droite de la Figure 1. Comme visible sur ladite image, les forces en question se divisent en deux en fonction de la masse sur laquelle elles s'appliquent :

➤ La première masse est soumise à :

- La gravité \vec{g}
- La force de frottement fluide \vec{f}_f
- La force de rappel du premier ressort \vec{R}_{10}
- La force de rappel du second ressort $-\vec{R}_{21}$

➤ La seconde masse est soumise à :

- La gravité \vec{g}
- La force de frottement fluide \vec{f}_f
- La force de rappel du deuxième ressort (entre les deux masses) \vec{R}_{21}

Même si le nombre de forces en présence semble conséquent, la réalité est toute autre et ce type de simulation a été rapidement mis en place. En effet les principales étapes étaient déjà calculées au préalable (le calcul des forces, l'intégration temporelle, ...) il ne restait donc plus qu'à adapter les équations déjà implémentées de façon à prendre en compte la nouvelle masse mais également de l'afficher. Notons ici tout de même l'importance du signe '-' dans le calcul de la force de rappel du second ressort sur la première masse, ce qui permet de ne calculer la force qu'une seule fois sur la seconde masse et de déduire la force sur la première masse par réciprocité.

Cette simulation nous permet alors d'arriver à des résultats cohérents (en ayant bien sûr choisi les bons paramètres au préalable pour les raisons mentionnées dans la section 2.2 précédente) comme le montre la succession d'images de la Figure 3. On peut en effet y deviner que le comportement des deux masses est bien différent d'un comportement de balancier simple de par le fait que la masse la plus basse donne une impulsion supplémentaire et en général de sens opposé à la masse centrale. Ceci est parfaitement caractéristique d'un double pendule et l'on retrouve des résultats satisfaisants. En effet ceux-ci ne sont pas et ne peuvent pas être exactement révélateurs de la réalité à cause de la discrétisation dont les problèmes et limitations ont été évoqués plus haut dans le rapport.



Figure 3: Illustration du bon fonctionnement du double ressort par une succession d'images prises les unes après les autres.

3 Simulation de tissu

Cette section vise à décrire le processus permettant de simuler le comportement d'un tissu en fonction des forces qui lui sont appliquées sans contraintes apparentes. L'ajout de contraintes sera développé dans la section 4. Notons d'ores et déjà qu'un tissu va être considéré par la suite comme une grille de points reliés entre eux par des ressorts comme illustré sur la Figure 4.

3.1 Intégration temporelle

Comme précédemment et pour des raisons évidentes de simulation numérique, nous sommes contraints de passer par une discrétisation de l'équation continue du mouvement régissant le comportement du tissu. Pour ce faire, une fonction dédiée à l'intégration temporelle est mise en place de façon à ce que l'actualisation des vitesses et des positions se fasse selon l'équation 4.

$$\begin{cases} v_{t+1} = (1 - \Delta t \cdot \mu) \cdot v_t + \Delta t \cdot \sum F_t \\ p_{t+1} = p_t + \Delta t \cdot v_t \end{cases} \quad (4)$$

avec v_t et p_t la vitesse et la position d'un point du tissu à l'instant t , Δt le pas d'intégration temporelle, μ le coefficient d'amortissement et $\sum F_t$ la somme de toutes les forces agissant sur le point du tissu à l'instant t .

Attention ici, nous avons longtemps eu des problèmes lors de la simulation qui suit parce que nous n'avions pas pris en compte la force de frottements fluides. Or notre tissu étant composé de ressorts, cette force est particulièrement importante pour la convergence de la simulation. En effet, sans cette force, nos ressorts auraient une oscillation perpétuelle (comme évoqué lors de la précédente section 2) ce qui impliquait donc que même lorsque notre tissu était censé être à son état d'équilibre, celui-ci continuait de trembler car les ressorts n'étaient jamais amortis et donc en permanence en train de s'attirer et de se repousser.

Néanmoins, au stade actuel de l'implémentation, sans considérer ces forces de rappel des ressorts, seule la gravité est implémentée sur le tissu et il tombe verticalement et parfaitement parallèle au sol.

3.2 Forces

Comme précédemment décrit, nous considérons chaque point du tissu comme étant relié à ses voisins par des ressorts. Attention cependant, la façon la plus naturelle de déterminer les voisins d'un point serait de prendre les voisins dits directs, c'est à dire ceux présent sur l'image de gauche de la Figure 4. Or si effectivement ces voisins sont nécessaires afin de déterminer la structure du tissu, ils ne sont pas suffisants. Afin de pouvoir simuler le plus fidèlement possible le comportement d'un tissu, il est également nécessaire de prendre en compte ses voisins diagonaux responsables de la force de cisaillement, ainsi que les voisins latéraux éloignés afin de simuler la force de courbure dont ils sont responsables. La Figure 4 permet de visualiser la position de ces dits voisins avec de gauche à droite les ressorts structurels (*Structural*), les ressorts de cisaillement

(*Shearing*) et les ressorts de courbure (*Bending*) précédemment décrits. Nous allons par la suite voir quelles sont les différences d'implémentation de chacun de ces types de ressorts, en appuyant plus sur les ressorts structuraux afin de montrer que ceux-ci ne sont pas suffisants pour modéliser le comportement d'un tissu.

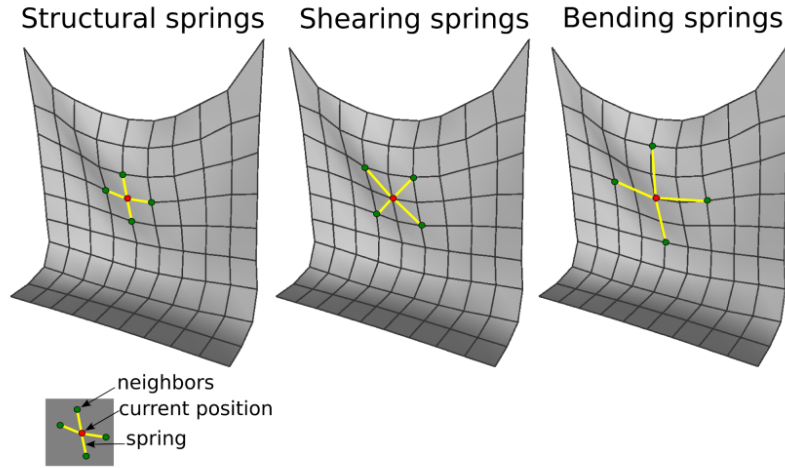


Figure 4: Schema décrivant les trois types de ressort qui s'appliquent en un point du tissu (point rouge).

Force de ressorts structuraux (*Structural*) La force de rappel des ressorts structuraux permet en effet de faire en sorte que le tissu reste un tissu durant toute la simulation et permet d'éviter entre autre des phénomènes engendrant des trous dans le tissu (voir Figure 5 lorsqu'une collision avec une sphère - développée en section 4 - est effective mais qu'aucune force de ressort n'est présente).

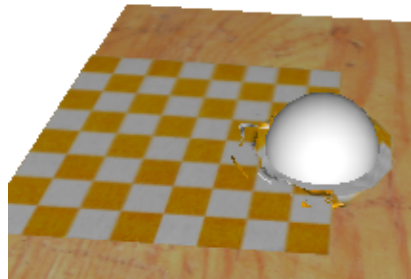


Figure 5: Formation d'un trou dans le tissu puisqu'aucune force n'est présente pour garantir la structure du tissu.

Cette force consiste donc à prendre chacun des voisins directs d'un point du tissu et de simuler le lien qu'ils ont avec le point courant par un ressort dont l'équation est déterminée par l'équation 2 avec $L_0 = 1/N$ avec N le nombre de points par ligne (et par colonne) puisque notre tissu a une dimension totale de 1×1 . Notons que si le nombre de lignes et de colonnes était différent, dans ce cas, il faudrait adapter la valeur de L_0 selon si l'on travaille sur un voisin de droite/gauche ou de haut/bas. Ainsi en n'utilisant que cette force structurale et en fixant deux points du tissu comme fixés à leur position d'origine il est possible de voir l'évolution du tissu comme illustrée en Figure 6.



Figure 6: Simulation du comportement d'un tissu tenu par deux extrémités et soumis uniquement à la gravité et aux forces des ressorts structuels.

On peut alors constater sur ces figures que le tissu se comporte bien comme attendu, l'ensemble des points sont ramenés vers le centre en cascade comme pour un vrai tissu, à une exception près : le tissu a tendance à s'allonger au fur et à mesure de la simulation.

Aussi essayons de modifier la constante de raideur de tous les ressorts en jeu ici. La figure 7 illustre la variation de la raideur de ces ressorts et son impact sur le tissu mais celui-ci ne se comporte pas comme espéré et semble particulièrement étiré dans les coins. De plus dès que le seuil de 50 est dépassé, la simulation commence à diverger de plus en plus rapidement. Nous nous trouvons donc là face à une impasse : les résultats ne sont pas satisfaisants et le paramètre permettant de rendre le tissu plus rigide est limité à cause de la discrétisation qui cause une divergence.

Ceci permet donc bien de montrer que seuls les ressorts structuels ne sont pas suffisants pour simuler un tissu. De plus dans la réalité il manque les deux forces mentionnées précédemment : la force de cisaillement qui empêche les points du tissu à trop se décaler les uns par rapport aux autres, ainsi que la force de courbure qui permet de modéliser comme son nom l'indique un meilleur effet de courbure du tissu.

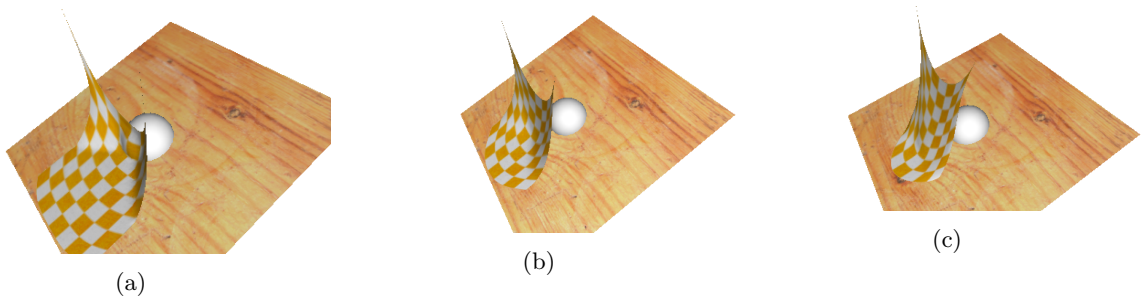


Figure 7: Comparaison du tissu en fonction de la raideur des ressorts de structure (de la droite vers la gauche la constante K_{struct} prend les valeurs 5, 20 et 50. On observe bien que plus K_{struct} est élevé plus le tissu est structuré et résiste mieux à la gravité. Pour un K_{struct} faible on peut remarquer que le tissu est étiré de façon non réaliste

Force complémentaires (*Shearing*) Afin d'améliorer le modèle du tissu, l'ajout de deux forces de ressort est nécessaire : la première est fondamentale et ajoute des ressorts liants les points par leurs voisins diagonaux comme décrit dans la figure 4. Cela a pour effet de limiter les déformations diagonales du tissu et de pouvoir espérer de garder une forme carrée au tissu.

Cette force de ressort dépend d'une raideur et en jouant avec le paramètre comme dans la figure 8 on montre bien que les déformations diagonales sont réduites et que cette force permet notamment de contrôler l'étirement du tissu dans les coins. Durant ce test, hormis les ressorts de cisaillement seule la force de structure est présente avec une raideur égale à 1.

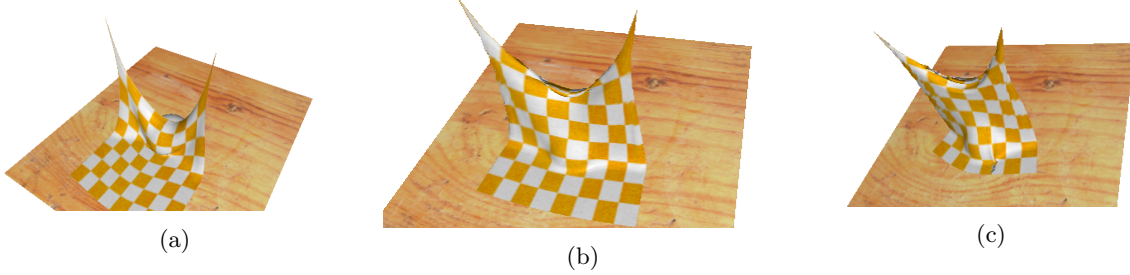


Figure 8: Comparaison du tissu en fonction de la raideur des ressorts de cisaillement (de la gauche vers la droite la constante K_{shear} prend les valeurs 1, 10 et 40. On observe que plus K_{shear} est élevé moins le tissu est déformé diagonalement.

Force de ressorts de courbure (*Bending*) Afin d'aller encore plus loin dans le modèle il peut y être ajouté des ressorts de courbure qui sont similaires aux ressorts de structure (figure 4) mais en permettant d'ajouter les voisins horizontaux et verticaux espacés de 2 sommets. Cette force permet de contrôler la courbure du tissu pour que celui-ci ne soit pas complètement tordu comme nous pouvons le voir dans la figure 9 qui permet de voir l'impact de ces ressorts. Durant ce test, hormis les ressorts de courbures seul la force de structure est présente avec une raideur égale à 5.

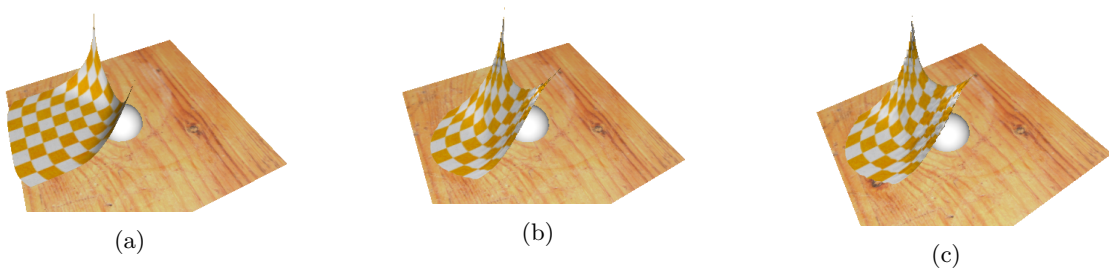


Figure 9: Comparaison du tissu en fonction de la raideur des ressorts de courbure (de la gauche vers la droite la constante K_{bend} prend les valeurs 2, 15 et 40. On observe que plus K_{bend} élevé moins le tissu pourra être "plié".

Connaissant l'impact de ces forces et leurs caractéristiques, il a été choisi en utilisant les trois types de ressort : $k_{struct} = 40$, $k_{struct} = 8$ et $k_{struct} = 8$ qui permettent un bon compromis sans trop rigidifier le tissu. Ces paramètres seront utilisés dans toute la suite du TP.

4 Gestion des contraintes

L'objectif est de pouvoir gérer les collisions du tissu avec les objets de la scène et de ne pas passer à travers. Les collisions étudiées sont une collision avec une surface plane et avec une sphère. La première étape pour gérer la collision est donc de pouvoir vérifier si un point entre en contact avec l'objet en question, pour la

surface plane il suffit de vérifier si les coordonnées de chaque point p dépassent le plan créé par la surface. Dans le TP la surface est orientée orthogonalement à l'axe z donc il suffit de vérifier la coordonnée z de chaque point p du tissu. De plus on peut noter que physiquement il aurait fallu vérifier que chaque point se trouve au-dessus de la surface (qui est un rectangle et non pas un plan) cependant dans notre cas le tissu ne va jamais en dehors de la surface donc cela n'a pas été implémenté. Ainsi quand un point atteint la surface il suffit de replacer sa position pour être écarté de la surface par un ϵ proche de zéro avec la formule 5.

$$Position(p.z) = z_{surface} + \epsilon \quad (5)$$

Pour la sphère, quelques calculs sont nécessaires car l'appartenance d'un point p dans une sphère se fait à l'aide de la distance du point par rapport au centre c de celle-ci (équation 6). De plus une fois qu'un point est dans la sphère il est nécessaire de calculer la direction normalisée $direction_{vec}$ du centre vers le point pour ensuite replacer le point à une distance de $r + \epsilon$ partant du centre c selon cette direction avec ϵ tendant vers zéro (équation 7).

$$\sqrt{(p.x - c.x)^2 + (p.y - c.y)^2 + (p.z - c.z)^2} \quad (6)$$

$$Position(p) = direction_{vec}(p) * (r + \epsilon) + c; \quad (7)$$

Cependant il se trouve qu'en modifiant seulement la position il implique que la vitesse et les forces des points sont conservées, cela va faire que les points vont retourner dans l'objet de collision et ainsi "rebondir" ce qui rendra instable la collision. En pratique il peut être observé que le tissu tremble sur la sphère ce qui n'est pas très réaliste. Pour ce problème il faut étudier physiquement l'influence de la collision sur l'objet et ainsi modifier aussi la vitesse et la force du point qui entre en collision.

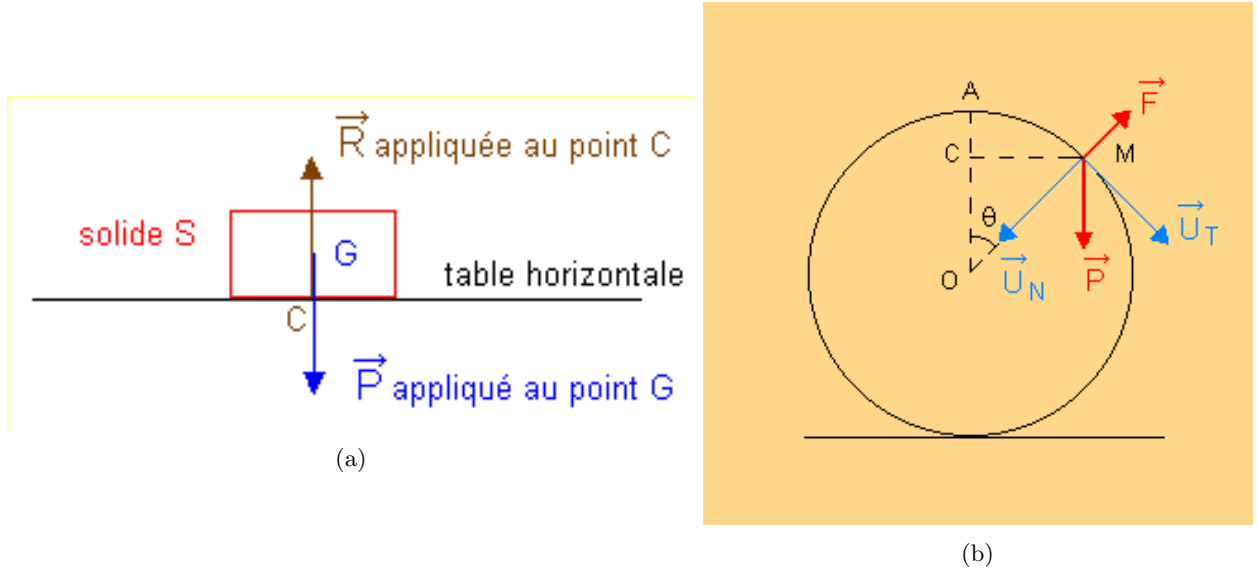


Figure 10: Schema des forces d'un point sur une surface plane et sur une sphère. A gauche les forces s'appliquent en un objet sur une surface plane et à droite les forces s'appliquent sur la sphère. Sur le schéma de la surface dans le cas particulier d'un point p les points C et G sont confondus sur p avec R la force de support de la surface en p . Sur le schéma de la sphère le point p est confondu avec M et avec F la force de support de la sphère en p . Dans les deux cas la force P représente la force de gravité

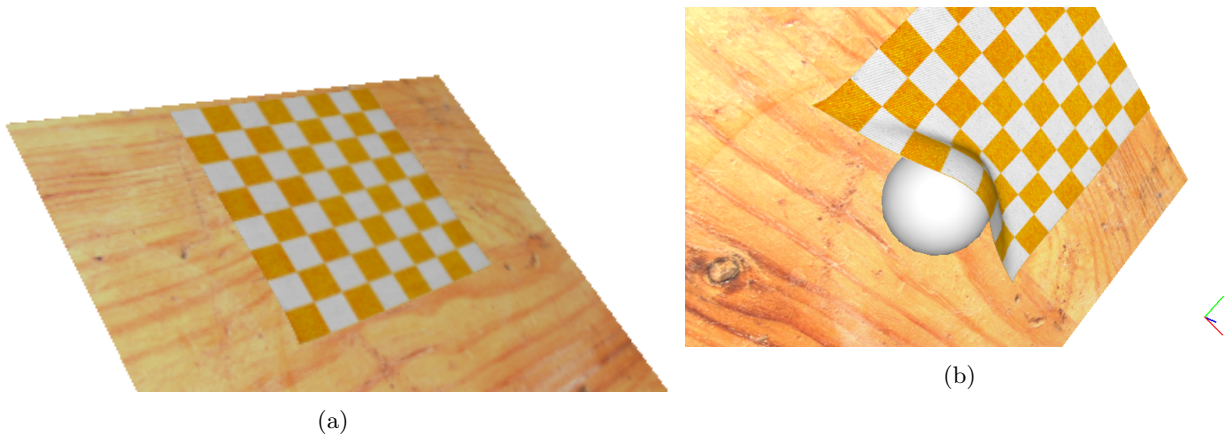


Figure 11: Images attestant du bon fonctionnement des collisions et de la stabilité de la position finale. à gauche la collision du tissu avec le plan et à droite la collision avec le plan et la sphère.

Pour la surface, selon le principe fondamental de la dynamique la force de support de la surface est opposé à la force de gravité ainsi la force suivant z est réduite à zéro (figure 10). En toute rigueur il faut en réalité mettre à zéro seulement si la force se dirige vers la surface, en effet il pourrait se produire une situation où le tissu serait soulevé par une force extérieure (une main) et mettre à zéros la force et la vitesse annulerai cette force sans raison physique.

Pour la collision de la sphère, son action est similaire à la surface plane à la différence que la force de support de la sphère est dirigée selon la direction du centre vers le point (figure 10). Il suffit donc de projeter les forces du point p sur la direction normale à la sphère en p et ainsi soustraire à la force du point p cette force projetée. Tout comme pour la surface il faut cependant vérifier l'orientation de la force par rapport à la sphère à l'aide du produit scalaire pour ne pas annuler des forces qui n'ont pas lieu d'être annulées comme il peut être illustré dans la figure 12.

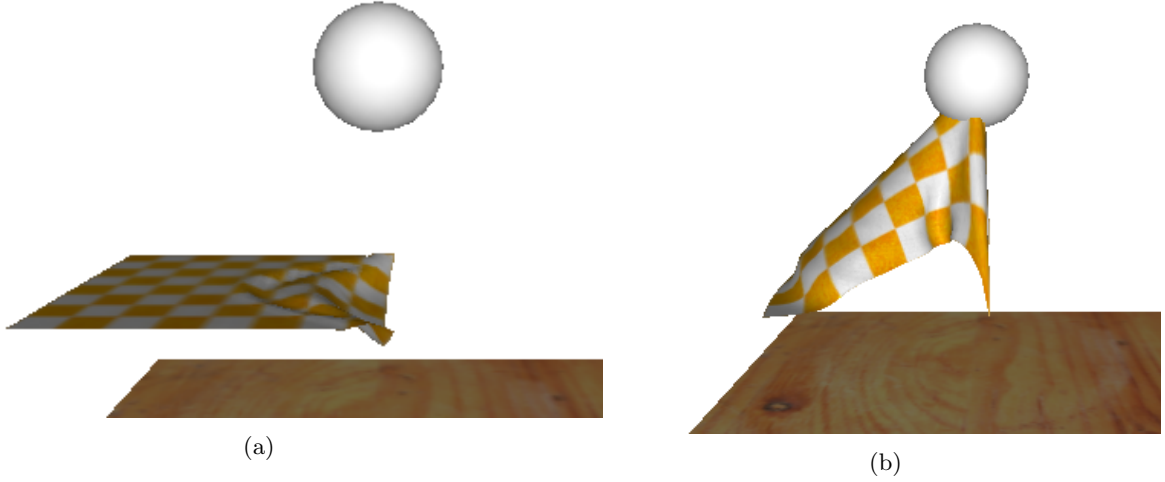


Figure 12: Comparaison de l'action de la gravité et de la gestion de collision entre le tissu et la sphère tel que la position initiale du tissu est dans la sphère. à gauche le résultat après un petit temps en ayant géré le sens de la force du poids et celle de la sphère, à droite le résultat après le même temps en annulant les forces sans tenir compte des directions. On peut visuellement voir que sans la gestion du sens des forces, la collision annule la force de descente ce qui résulte en le fait que le tissu est "collé" à la sphère.

Afin de pouvoir aller plus loin dans la simulation il est possible d'ajouter une autre contrainte qui est une force de vent qui s'applique sur le tissu. Cela permet ainsi d'observer comment celui-ci réagit et d'autant plus valider le modèle. La force de vent dépend d'une direction u_w (équation 8) ainsi que de la normale n qui est obtenu par le mesh du tissu et aussi d'une constante K_w permettant de contrôler la puissance du vent.

$$F_w = K_w \langle n, u_w \rangle n \quad (8)$$

En modifiant la coefficient K_w on peut vérifier que la simulation est cohérente avec la réalité (figure 13)

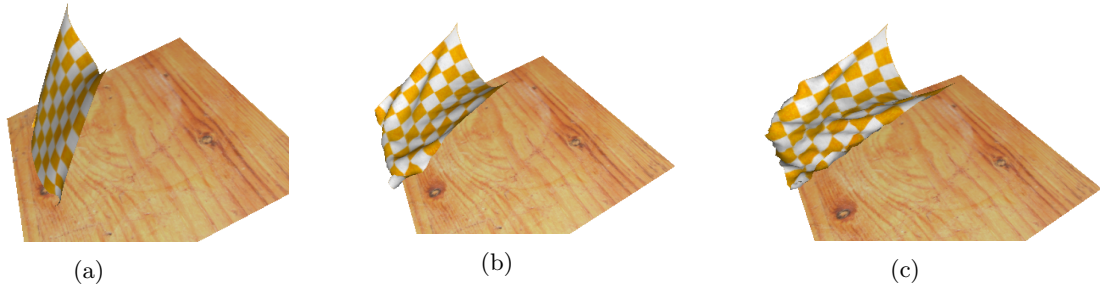


Figure 13: Comparaison du comportement du tissu en fonction de la puissance du vent (de la gauche vers la droite la constante K_w prend les valeurs 0.01, 0.02 et 0.05. On observe que plus le vent est élevé plus le tissu est poussé et agité ce qui est tout à fait réaliste.

On observe donc que le modèle de ressort reste tout à fait réaliste et stable avec des contraintes tel que des collisions ou une force de vent.

5 Implémentation GPU avec CS

Une fois que tout fonctionne sur CPU, il est commun de vouloir utiliser le GPU pour ses capacités de calcul. D'autant plus que le problème est facilement parallélisable du fait que nous calculons des forces en différents points. La solution proposée est d'utiliser 2 compute shader qui seront tout 2 utilisés sur $N*N$ groupes de processeur. Cela aura donc pour but que chaque groupe travaillera sur un point du tissu en particulier choisi à l'aide des identifiants du groupe de processeur obtenu avec `"glGlobalInvocationID.xy"`. Tout d'abord 4 buffers sont créés qui sont respectivement les positions des points du tissu, les vitesses des points, les forces des points et enfin les coordonnées de texture qui seront utilisées pour la texture du tissu. Il est aussi notable que les différents buffers soient des vec4 contrairement à la version GPU. Cela est simplement dû à l'alignement des données et que la 4eme composante sera mise à 0 et ne sera pas utilisé pour l'affichage final. Le premier compute shader `"force.cs"` est chargé de mettre à jour la force qui s'applique en chaque point. Les calculs de force étant identiques à ceux de la première partie du TP sur CPU elles ne seront pas détaillées ici. Il faut cependant noter que comme les forces de ressorts se calculent en fonction des positions il faut bien donner en entrée du compute shader le buffer de positions. Le deuxième shader `"integration_step.cs"` est lui chargé de mettre à jour les positions et la vitesse des points en fonction des nouvelles forces préalablement calculées. Ainsi il est nécessaire pour ce shader de recevoir les buffers de positions et de vitesses pour pouvoir les mettre à jour, ainsi que le buffer de force car la mise à jour dépend des forces.

Il faut cependant noter que pour la force du vent, la normale du point p n'est plus disponible dans le mesh et donc dans le compute shader. Il est donc nécessaire de créer les normales en utilisant les voisins du point p . En effet la normal peut être obtenue en réalisant le produit vectoriel entre les deux vecteurs partant de deux voisins allant vers p . une fois cette normale calculée il suffit de faire comme il a été fait sur le CPU et de calculer la force correspondante à la formule.

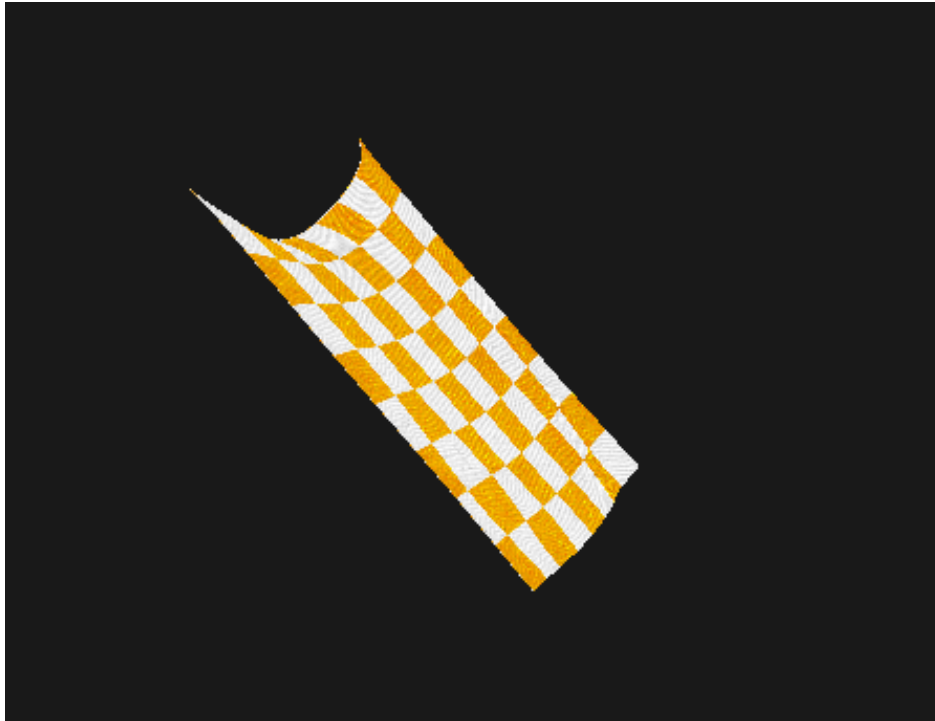


Figure 14: Images attestant du bon fonctionnement de la simulation du tissu avec les trois forces de ressort et la force du vent sur GPU avec les compute shaders.

Du fait de la rapidité d'exécution du code on peut s'amuser à soumettre le tissu à un vent puissant et le

voir s'agiter pour encore une fois visualiser la cohérence et le réalisme du modèle (figure 15).

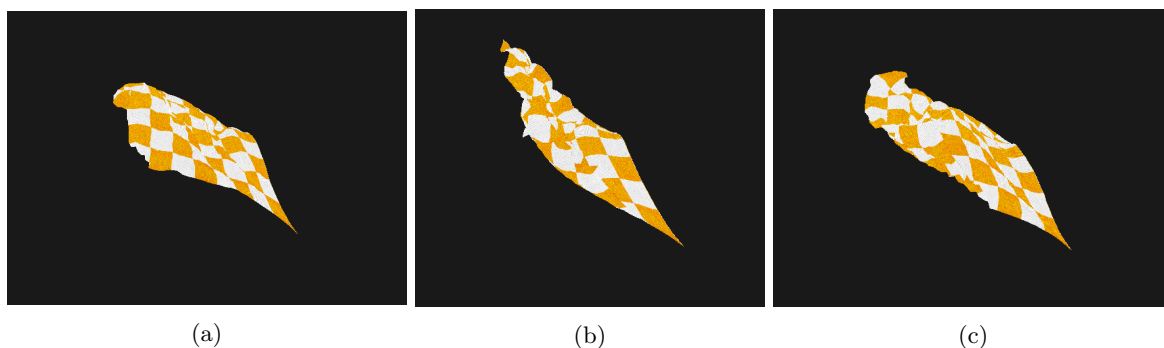


Figure 15: Images de l'animation du tissu avec un fort vent, l'animation se comporte exactement comme un réel tissu.

6 Conclusion

La modélisation informatique peut permettre de simuler et prédire les événements physiques qui peuvent être divers allant de la météo, les inondations, ou la simulation de l'eau. La modélisation peut aussi servir à vérifier des théories. Durant ce TP, il a pu être montré que l'animation informatique d'un tissu peut s'approximer comme un ensemble de points liés avec différents ressorts et les résultats semblent montrer que cette modélisation semble cohérente à condition de bien vérifier les lois physiques. De plus on note que la simulation informatique peut rapidement représenter beaucoup de calculs à faire du fait que la simulation se ramène à une échelle de particule, cependant le fait d'utiliser des particules ou une petite échelle est très facilement parallélisable et il est possible de pleinement utiliser la puissance de calcul du GPU. Notons finalement que la simulation numérique reste toujours qu'une approximation de la réalité du fait de la nécessité de discrétiser le temps continu. Ceci implique donc la possibilité de nombreux artefacts ou comportements inhabituels et non réalistes de certaines simulations voire jusqu'à la non stabilité de la simulation.