



Université Nationale de Colombie



*Université Claude Bernard Lyon1-
LIRIS-Equipe SAARA*

RAPPORT DE STAGE

Projet SAGA : Simulateurs pour l'Apprentissage des Gestes de l'Accouchement

Auteur :

Yury ESTEPA-
AVELLANEDA

Responsables :

Fabrice JAILLET
Jan BACCA-RODRIGUEZ

Avril - Octobre 2016

Table des matières

1	Introduction	1
	Introduction	1
2	Traitement des fichiers du maillage à NURBS	3
2.1	3DS Max 2015	3
2.1.1	Aménagement du maillage	3
2.1.2	Méthode 3DSMax	9
2.1.3	Difficultés dans la conversion en utilisant 3dsMax	12
2.2	Rhinoceros 5.0	13
2.2.1	Aménagement du maillage	13
2.2.2	Méthode en utilisant Rhinoresurf	17
2.2.3	Méthode en utilisant Cage Edit	22
2.2.4	Difficultés dans la conversion en utilisant Rhinoceros	25
3	Generation des tétraèdres à partir du maillage	26
3.1	Meshlab	26
3.1.1	Aménagement du maillage	26
3.1.2	Résolution des problèmes	27
3.1.3	Exportation du maillage à format .stl	28
3.2	Netgen 5.0	28
3.2.1	Conditionnement du maillage, format .stl :	28
3.2.2	Résolution des problèmes	28
3.2.3	Génération des tétraèdres à partir du maillage :	30
3.2.4	Exporter du maillage en utilisant le format gmsh2	33
3.2.5	Caractéristiques du format gmsh2	35
4	Detection des pressions exercées sur la tête foetale	36
4.1	Forceps	36
4.1.1	Sélection des forceps	38

4.2	Capteurs de pression	38
4.2.1	Sélection des capteurs	38
4.2.2	Caractéristiques des capteurs sélectionnés	39
4.2.3	Conditionnement des capteurs	40
4.2.4	Calibrage des capteurs	47
4.2.5	Mise en place des capteurs de pression sur le forceps	48
4.3	Capteur d'écartement	50
4.3.1	Mise en place de capteur	50
4.4	Récupération et traitement des données :	50
4.4.1	Caractéristiques de la Carte Arduino	50
4.4.2	Connexion des Capteurs de pression sur la carte Arduino	51
4.4.3	Connexion de Capteur d'écartement sur la carte Arduino	51
4.4.4	Codage sur Arduino	55
4.5	Interface Graphique	56
5	Conclusions et travaux futurs	57
6	Annexes	58
6.1	Annexe A. Code Arduino	58
6.2	Annexe B. Code Processing	62
6.3	Annexes C. Carte de Prototypage	66
6.4	Annexes D. Couplage Dspace Birthsim	67
	Références	71

1 Introduction

L'accouchement est l'action de mettre un enfant au monde. Il y a des étapes pour faire l'accouchement. Parmi elles, la phase de la dilatation, de l'expulsion et de délivrance [1]. Pendant ces étapes, il faut que les procédures soient bien appliquées par le personnel de la santé, parce que toutes les procédures impliquent des risques de la vie de la mère ou parturiente et de la vie du bébé.

Pour cette raison, divers simulateurs numériques ont été créés afin de donner la possibilité de recréer les étapes de l'accouchement et de cette façon, servir comme un outil d'apprentissage pour les étudiantes des domaines de la santé [2].

Le premier logiciel a été développé par l'Université de Stanford [3], ce logiciel a été le premier à mettre en application la réanimation néonatale. Après, différents logiciels ont été construits afin de représenter les signes vitaux [4], et les premiers simulateurs physiques ont été développé par SIMA, en cherchant représenter de manière complète la réanimation néonatale [5]. L'entreprise *Laerdal*, après le développement des simulateurs d'adultes [6, 7], a décidé de travailler sur la conception d'un simulateur de nouveau-né, capable de simuler les signes vitaux et d'autres conditions cliniques des nouveaux-nés[8].

En 2012, le projet **SAGA** (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement) a été créé. Ce projet est dédié au développement des outils pédagogiques, afin de simuler une ambiance complète de l'accouchement d'un nouveau-né. Cet projet a été divisé dans trois parties : *Modèle numérique*, *Système Physique* et une *Interface graphique*. Chacune partie aborde le projet en utilisant différents outils. Dans la figure 1, vous pouvez voir l'architecture du projet SAGA.

L'idée principale est d'utiliser ce système depuis une composante pédagogique, en permettant de designer des scénarios cliniques selon le public, les besoins et la difficulté souhaitée. [9]. Dans le schéma de la figure 1, on peut observer une relation entre chaque module, c'est pourquoi le système donne la possibilité de prendre en compte les actions implémentées dans d'autres parties en facilitant l'apprentissage et la vérification de chaque procédure développée.

Dans ce document, on montrera, en premier lieu, le traitement de maillages de la tête foetale, en utilisant des surfaces NURBS (non-uniform rational B-spline). Après, on va travailler sur la génération des tétraèdres à partir du maillage, afin d'inclure la tête dans la simulation numérique. De cette manière, pouvoir simuler en tenant en compte du volumen et de la déformation de la tête foetale en passant pour le canal d'accouchement. Ensuite, on montrera la mise en place de capteurs pour déterminer les actions et les forces exercées par les mains ou les forceps sur la tête foetale du simulateur haptique.

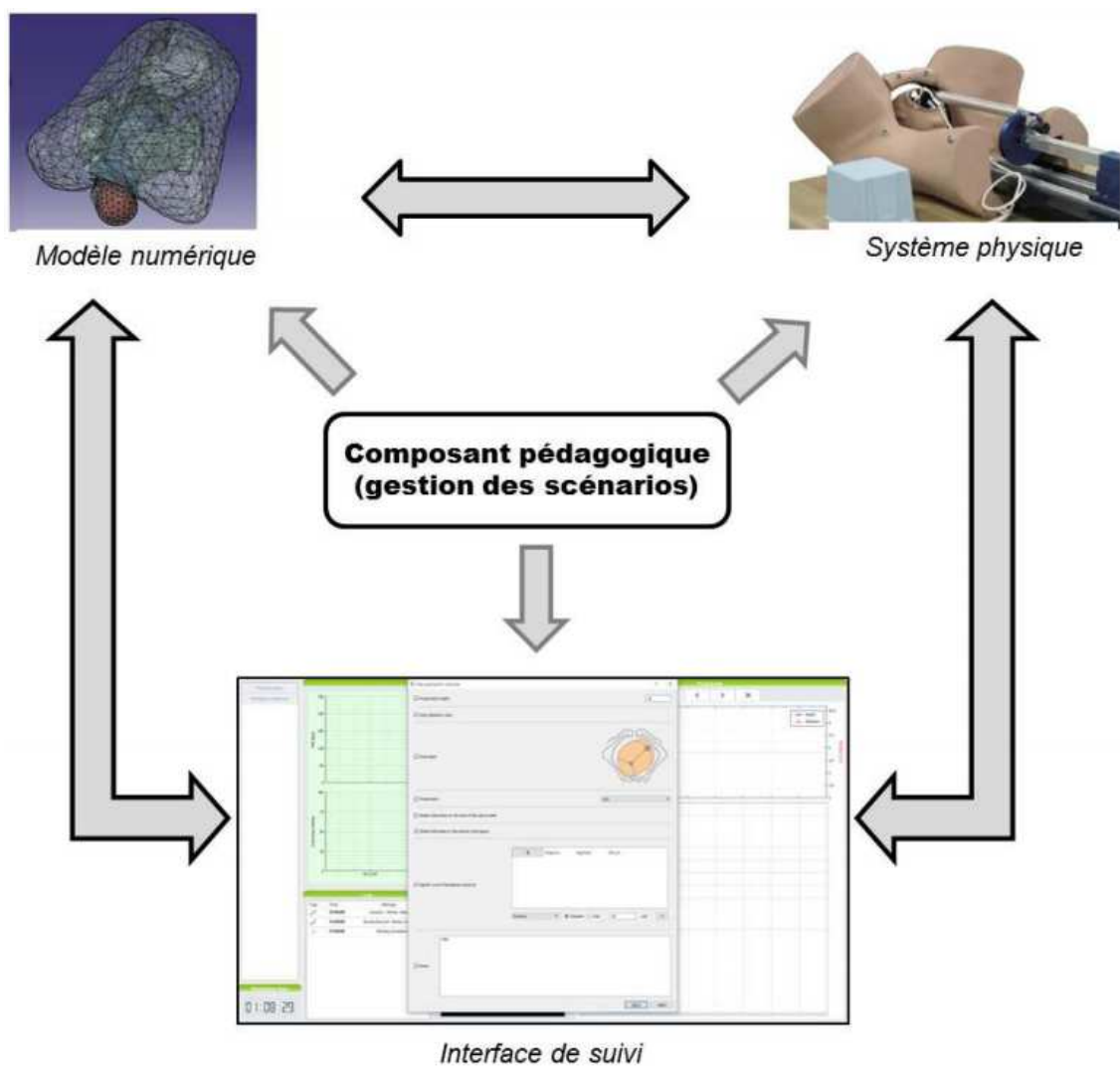


Figure 1 – Architecture du projet SAGA [9].

2 Traitement des fichiers du maillage à NURBS

Le projet SAGA, spécifiquement la partie liée au modèle numérique travaille avec des fichiers du maillage de la tête foetale qui sont utilisés afin de simuler l'structure physique et de représenter divers comportements physiques. En vue de cela, nous avons besoin d'obtenir un maillage propre et facile à utiliser. De même, l'idée est de faire une conversion à tétraèdres et de cette manière, simuler le volume et la déformation de la tête foetale en passant par le canal d'accouchement.

Pour cette raison, dans ce chapitre, on va montrer différentes méthodes de traitement des images en utilisant les logiciels 3DSMax et Rhinoceros et les difficultés trouvées avec chaque méthode.

2.1 3DS Max 2015

Ici, on va montrer le processus nécessaire pour la conversion des fichiers avec les extensions *.obj* et *.wrl*, en utilisant le logiciel *3DSMax Studio 2015*. (Les fichiers *.msh* ne peuvent pas s'ouvrir avec 3DSMax).

3DSMax sera utilisée avec une licence étudiante avec une validité de trois ans. De cette manière, on pourra travailler sans soucis. D'autre côté, 3DSMax est seulement disponible sur le système d'exploitation Windows.

2.1.1 Aménagement du maillage

1. *Ouverture Fichier .WRL :*

En premier lieu, on doit ouvrir 3DSMax. Ensuite on doit choisir l'option Import dans le menu qui se trouve dans la partie supérieure à gauche, comme le montre la figure 2 ci-dessous. Immédiatement, on doit sélectionner Import (*Non – native file formats into 3ds Max*). Cette action permet de sélectionner le fichier à importer avec les extensions *.obj* et *.wrl*, entre autres options.

On a utilisé le fichier appelé *foetal – head – 4466.wrl* pour montrer la procédure suivie. Nous chargeons le fichier et une fenêtre apparaît, comme le montre la figure 3.

Il y a trois options :

- **Reset scene** : Cette option efface les scènes antérieures, afin d'éviter le mélange avec le fichier à importer.
- **Turn to 3DS Coordinates** : Si nous choisissons cette option, on utilisera les coordonnées natives du 3DSMax, sinon, le logiciel maintiendra les coordonnées du fichier original.
- **Create Primitives** : Cet outil permet d'utiliser les fonctions primitives du 3DSMax et aussi, de prendre les textures. Sinon, 3DSMax transformera l'objet dans un

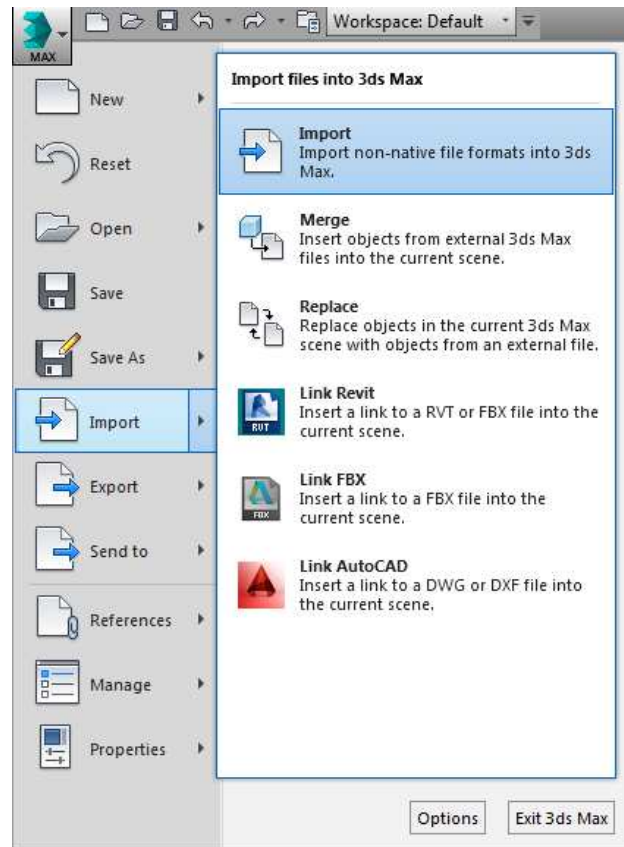


Figure 2 – Importer des fichiers en utilisant 3dsMax 2015.

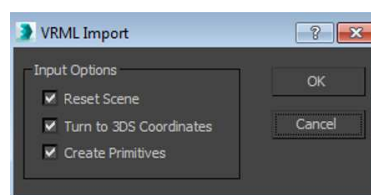


Figure 3 – Option pour importer un fichier .wrl dans le logiciel 3DSMax 2015.

maillage triangulaire.

On a laissé toutes les options sélectionnées et finalement on a choisi **OK**.

3DSMax donne une fenêtre avec quatre sous-fenêtres (Figure 4) qui permettent de travailler avec l'objet dans différents vues et perspectives.

2. Résolution : Nombre de polygones et triangles :

Il faut que nous observions le nombre de polygones et triangles que nous avons, afin d'analyser si cela est nécessaire de changer la résolution de notre objet. On peut voir les statistiques que montre 3DS Max 2015. On choisit le symbole + dans n'importe quelle fenêtre. Ensuite, nous cherchons l'option *Xview*–*Show Statistics*. Et on pourra

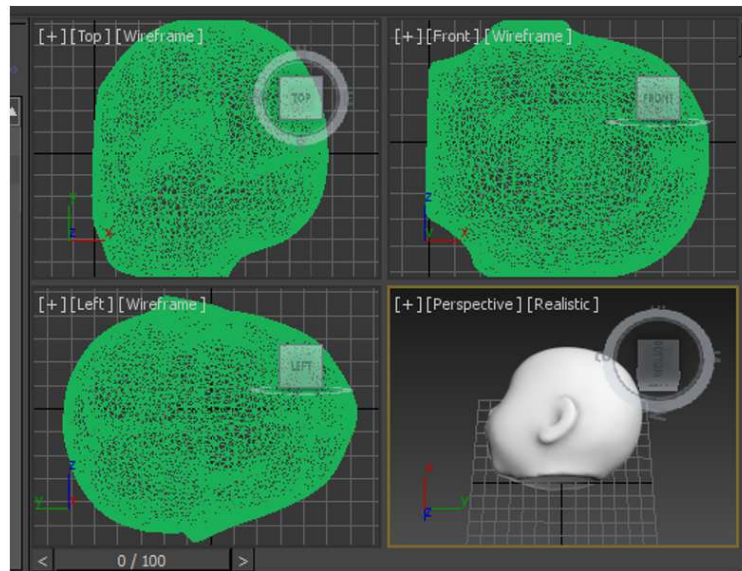


Figure 4 – Espace de travail dans le logiciel 3DSMax 2015.

voir le numéro des polygones, triangles et vertex, comme le montre la figure 5.

Alors, il est fondamental que nous travaillions avec moins de vertex et qu'on réduise

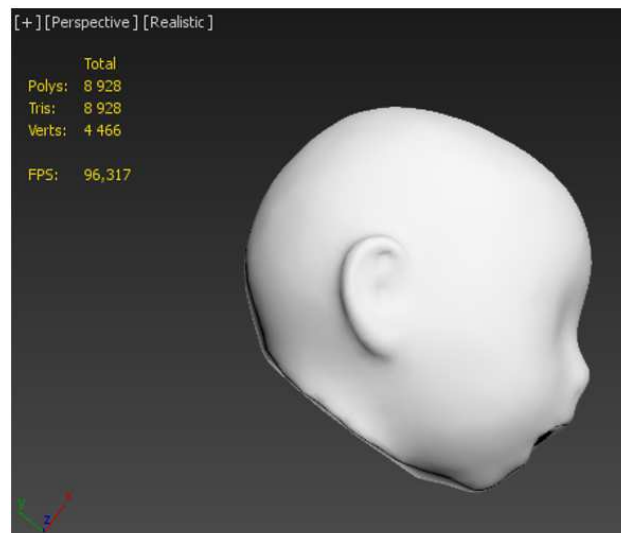


Figure 5 – Vue Perspective avec le numéro de polygones, triangles et vertex.

le nombre de polygones ou triangles. Pour cette raison, on va utiliser l'outil *Multires*. En premier lieu, on doit sélectionner la tête et sur le côté droit, on pourra voir une fenêtre avec le nom de l'objet et le type du format de l'image. Dans notre cas, il s'agit d'un maillage éditable. Après, dans la partie *Modifier List*, il y aura une liste déroulante, on doit chercher *Multires*, illustrée par la figure 6.

L'outil *Multires* montrera une liste des paramètres à modifier en fonction des besoins.

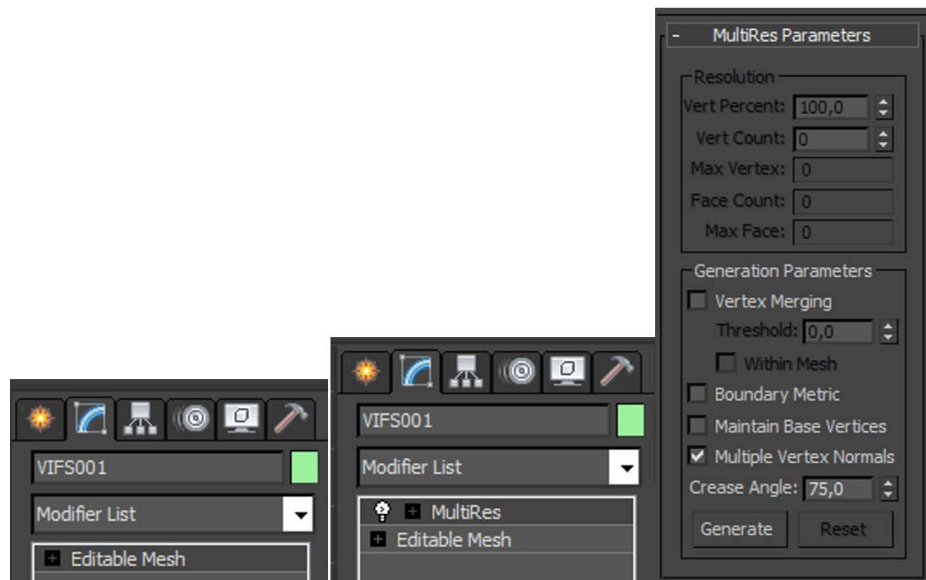


Figure 6 – Modificateur Multires dans 3DSMax 2015

Par rapport aux paramètres de la résolution, il montre les variables suivantes :

- **Vert Percent** : Correspond au nombre des vertex de l'objet modifié par rapport au nombre total de vertex du maillage d'origine.
- **Vert Count** : C'est-à-dire le numéro total de vertex de l'objet modifié.
- **Max Vertex** : Permet de voir le nombre des vertex du maillage original auquel nous appliquons *Multires*.
- **Face Count** : Montre le nombre de faces.
- **Max Face** : Indique le nombre maximal de faces.

Ces valeurs sont automatiquement mises à jour, quand nous changeons un paramètre. D'autre part, dans la section de génération des paramètres, on peut trouver les options suivantes :

- **Vertex Margin** : Si l'option est activée, *Multires* va fusionner les vertex de tous les objets que composent le maillage.
- **Boundary Metric** : Cette option permet de mélanger les frontières des toutes les éléments adjacents qui composent l'objet.
- **Maintain Base Vertices** : Cet outil permet de conserver les vertex qui sont la base de l'objet, afin de maintenir la structure, même quand la résolution est diminuée. Il est très important de savoir qu'après de la diminution de la résolution, les objets peuvent être méconnaissables.
- **Multiple Vertex Normals** : Il permet la génération de plusieurs normales. Elles sont appliquées dynamiquement comme les changements de résolution de sommet.
- **Crease Angle** : Il permet une variation de l'angle des normales entre 0 et 180. Et seulement, il est accessible si l'option antérieure est activée.

Finalement, on peut choisir l'option **Generate**, afin de générer l'objet avec les para-

mètres que nous avons choisi.

Pour les exemples suivants, on a sélectionné quelques vertex, afin de garantir la réduction dans ces zones. Dans la figure 7, vous pouvez voir le processus de réduction de polygones.

Dans la figure 8, on présente le nez et les oreilles sélectionnées comme vertex base.

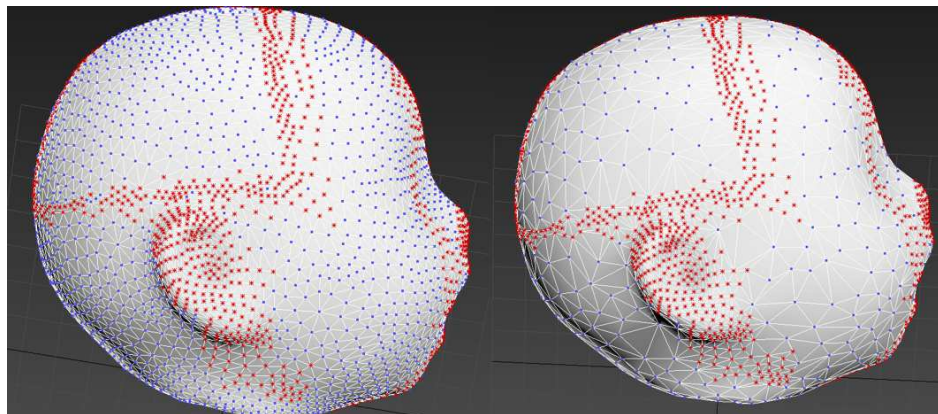
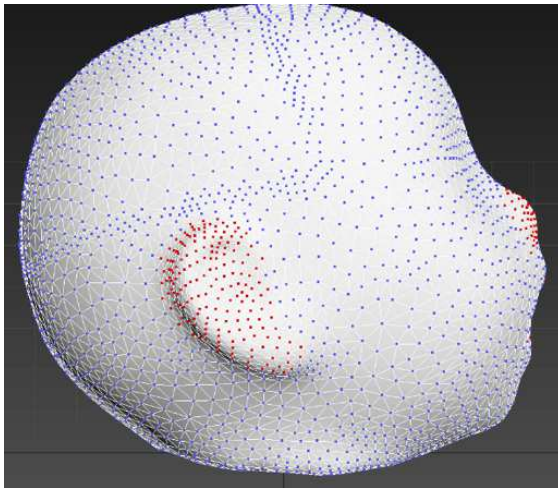
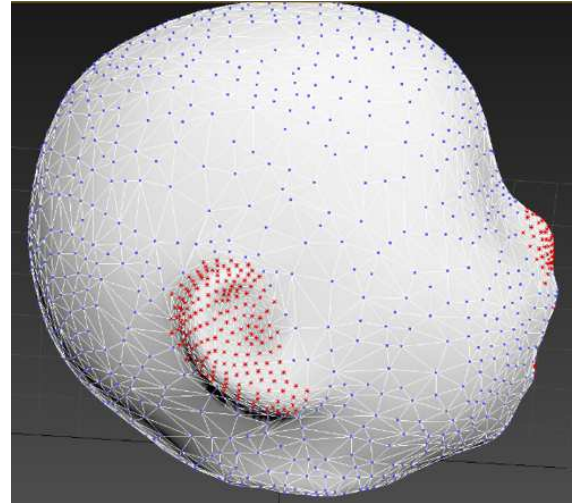


Figure 7 – Gauche : tête originale avec les fontanelles et les oreilles sélectionnés. Droit : tête avec une réduction de 50 pour cent de polygones.

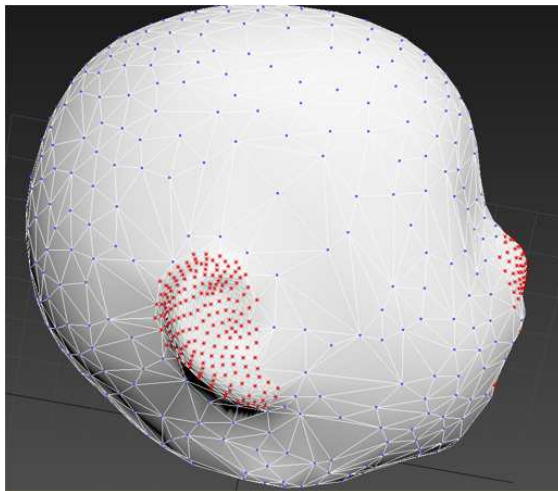
Après la réduction en polygones, c'est nécessaire de garder les changements faits. En premier lieu, on doit combiner notre objet **Editable Mesh** avec l'outil *Multires*. Pour cette raison, on utilise l'option **Collapse All** (Figure 9) . On a fait click droit, sur *Multires* et on choisit l'option **Collapse All**.



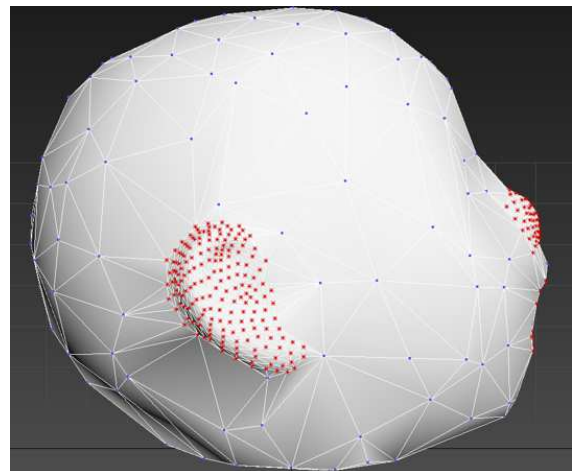
(a) Tête originale avec le nez et les oreilles sélectionnées.



(b) Tête avec une réduction de 50 pour cent de polygones



(c) Tête avec une réduction de 25 pour cent de polygones



(d) Tête avec une réduction de 15 pour cent de polygones

Figure 8 – Différentes reductions en utilisant l'outil Multires et Vertex base

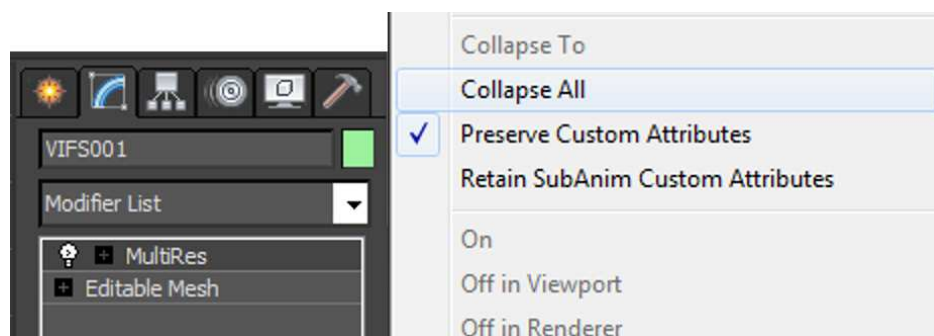


Figure 9 – Menu Collapse All dans le logiciel 3DS Max 2015.

2.1.2 Méthode 3DSMax

Le but final est de faire une conversion de format *.obj*, *.wrl* a *.dae*, afin de pouvoir travailler avec surfaces NURBS et de générer des tétraèdres. On va commencer en suivant les pas suivants qui se montrent, en utilisant le logiciel 3DSMax [10], afin d'obtenir un fichier ouvrable dans n'importe quel logiciel Collada Editor (Utilise le format *.dae*).

1. *Conversion Editable Mesh -> Editable Patch :*

On va garder le fichier en format *.max* (3DSMax). Ensuite, on fera un changement dans le type d'objet. On doit rappeler que nous sommes en train d'utiliser un format type *Editable mesh*. Et pour nos besoins finaux, on doit faire une conversion au format *Editable Patch*.

On va sélectionner *Editable Mesh*, on fait click droit et après (Figure 10), on peut choisir l'option *Editable Patch*.

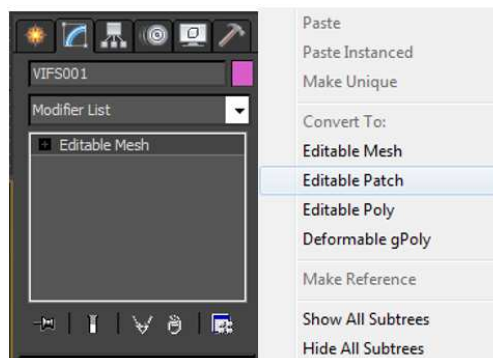


Figure 10 – Conversion Editable Mesh to Editable Patch dans le logiciel 3DS Max 2015.

2. *Conversion Editable Patch : -> NURBS*

Finalement, on fait la même procédure, pour effectuer le changement jusqu'au format NURBS. On va sélectionner *Editable Patch*, on fait click droit et après, on peut choisir l'option *NURBS*, comme le montre la figure 11. Dans la figure 12, se montre la tête foetale dans le format NURBS. Pour obtenir cette figure, il faut que le nombre de polygones soit le plus petit possible afin d'éviter que le logiciel 3DSMax ne soit pas capable de faire la conversion NURBS.

3. *Exporter le fichier en format .dae (OPENCOLLADA)*

Maintenant, on peut exporter notre tête foetale afin d'utiliser le modèle dans un environnement haptique. Pour cette raison, nous allons exporter en utilisant le format *OPEN COLLADA .dae*.

En premier lieu, on doit télécharger l'outil que se trouve dans la page officielle (<http://www.opencollada.org/>). Cet outil permettra que 3dsMax garde nos archives de manière correcte. Nous devons télécharger le fichier ColladaMaxNew.dle

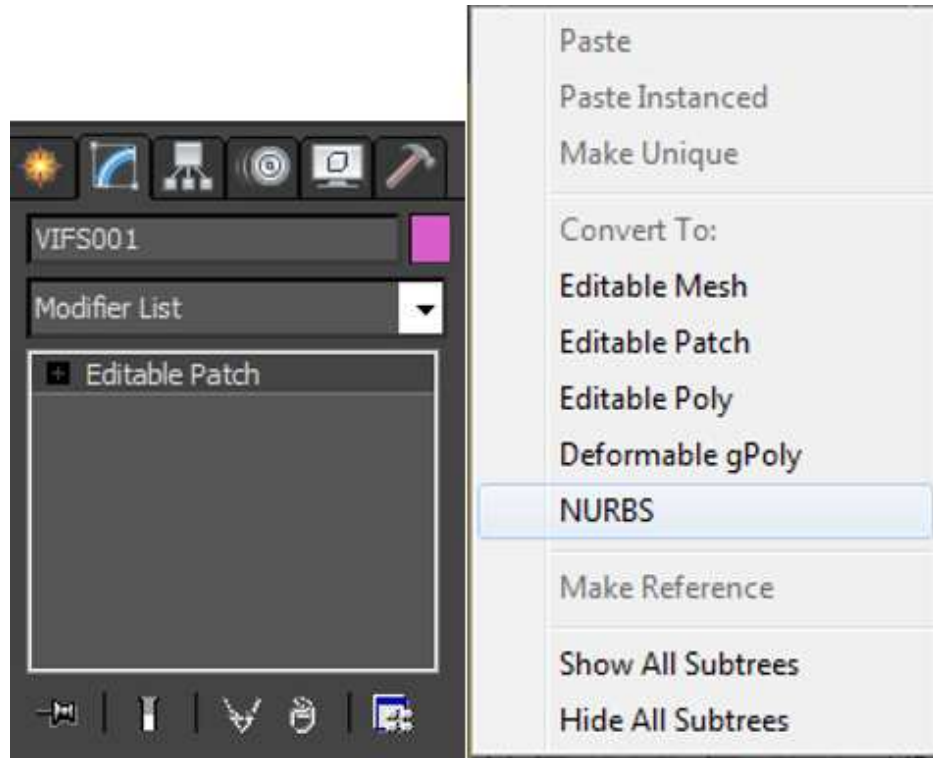


Figure 11 – Conversion Editable Patch to NURBS dans le logiciel 3DS Max 2015.

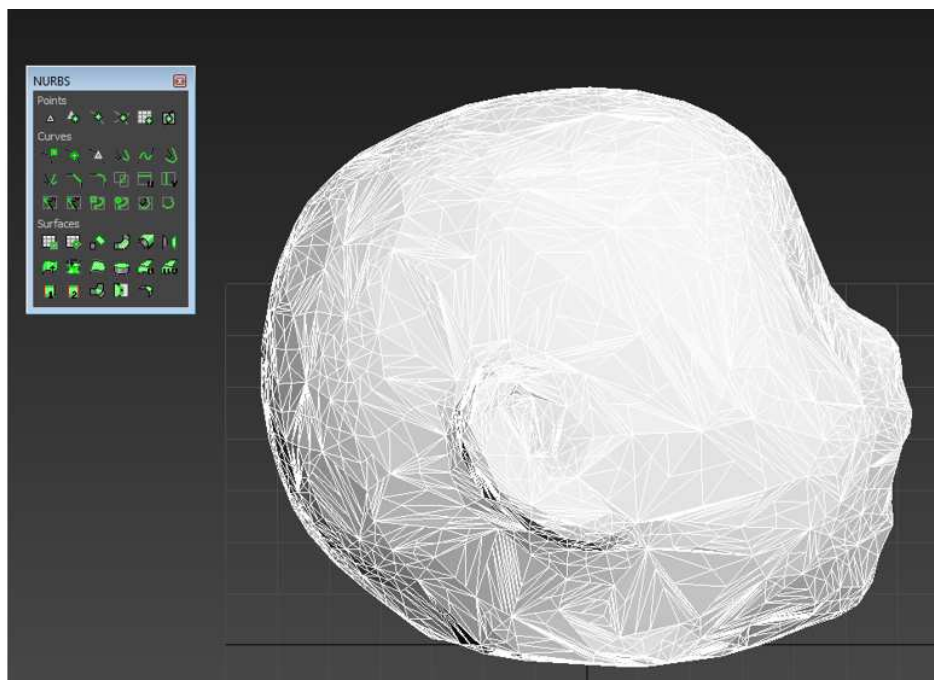


Figure 12 – Tête foetale dans le format NURBS.

(qui marchera seulement sur Windows), et on doit garder l'archive dans le dossier «Plugins». La fois suivante que nous utilisons 3ds Max, l'outil collada sera disponible.

Pour faire l'exportation du fichier, simplement on doit aller au menu principal et choisir l'option *Exporter-(Non-native file formats from the current 3ds Max Scene)*.(Figure 13)

3DS Max nous montrera une fenêtre des paramètres afin de bien exporter notre objet.

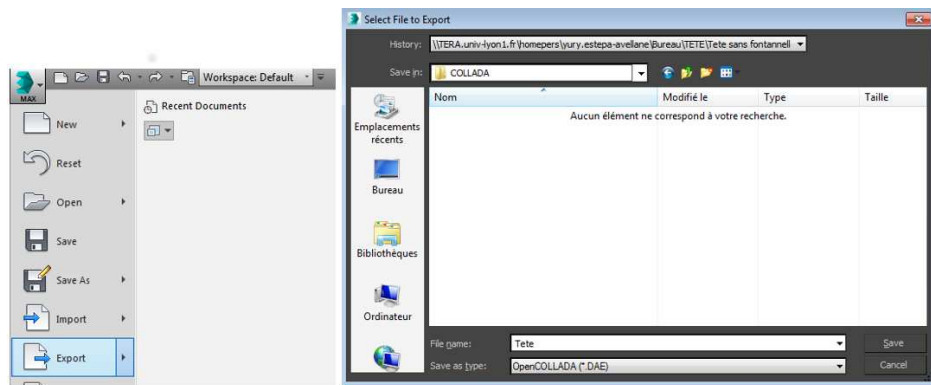


Figure 13 – Exporter des fichiers au format OPEN Collada en utilisant 3dsMax 2015.

Les paramètres se montrent dans la figure 14.

Il s'agit des trois zones :

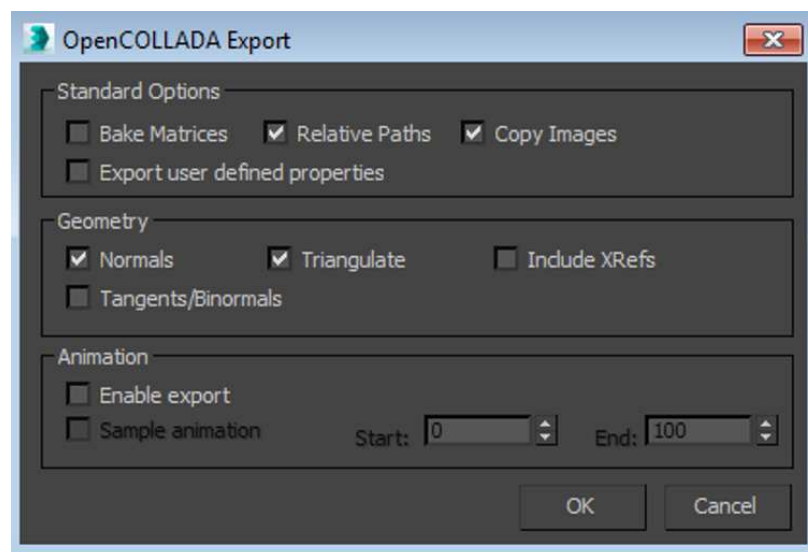


Figure 14 – Paramètres d'exportation OPEN Collada en utilisant 3dsMax 2015.

- **Standards Options** : Dans la première partie, nous trouvons quatre options :
 - *Bake Matrices* : Cet option permet la transformation de chaque noeud de la scène et de leurs animations que seront exportées comme matrices.

- *Relative Paths* : En utilisant cette propriété, on peut garder les images liées aux textures en utilisant un chemin relatif, afin d'éviter les soucis.
- *Copy Images* : copiera tous les fichiers de texture dans le dossier « Images » au-dessus de votre dossier de *.DAE*.
- *Export* : User defined properties.
- **Geometry** : Ici, on observe quatre options, aussi :
 - *Normals* : Permet de faire l'exportation des normales du maillage.
 - *Triangulate* : Permet la division du maillage en triangles.
 - Include XRefs.
 - *Tangents/Binormals* : Exportez les tangentes de géométrie et binormals pour toutes les données du maillage, aussi bien que leurs indices dans la réduction en mosaïque de maille.
- **Animation** : Dans cette zone, il y a la possibilité de gérer l'animation de la scène et de faire l'exportation.

Dans la figure 15, vous pouvez voir la tête foetale en utilisant un logiciel qui permet de voir les fichiers avec extension *.dae*.

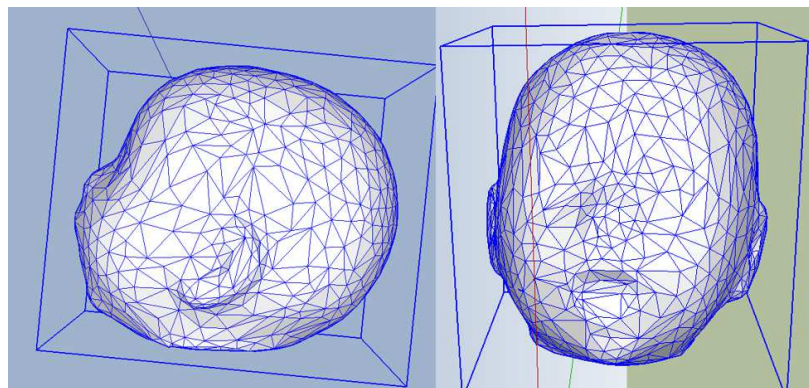


Figure 15 – Affichage tête foetale dans le format dae

2.1.3 Difficultés dans la conversion en utilisant 3dsMax

En utilisant le logiciel 3DS Max 2015 d'Autodesk, on peut travailler de manière facile quand on utilise des maillages polygonaux. Cependant, l'outil de conversion de maillages polygonaux afin d'obtenir NURBS n'est pas bon.

Ici, on montre les principales inconvenients :

- On doit réduire le nombre de polygones et même il y a des occasions où la tête a perdu ses caractéristiques physiques.
- 3DS Max ne peut pas créer de NURBS à partir du maillage, par contre, on doit faire une conversion a *Editable Patch* et ce processus complique le traitement du maillage.

- Le NURBS que génère 3DS Max n'est pas exploitable. Il génère un maillage en mosaïque à partir des polygones originaux. Finalement, nous avons beaucoup de points à travailler.

2.2 Rhinoceros 5.0

Ensuite, on va montrer le processus nécessaire afin de réaliser la conversion des fichiers avec les extensions *.obj* et *.stl* en utilisant le logiciel Rhinoceros 5.0 2015. Il y a la possibilité de télécharger une version trial de 90 jours pour ce logiciel. De même, le logiciel est disponible sur le système d'exploitation Windows et Mac.

2.2.1 Aménagement du maillage

1. Ouverture Fichier *.STL*

En premier lieu, on doit ouvrir Rhinoceros 5.0. Ensuite on doit choisir l'option Import dans le menu qui se trouve dans la partie supérieure à gauche, comme le montre la figure 16 ci-dessous.

Par la suite, on doit choisir le format du fichier à importer. Rhino permet de travailler

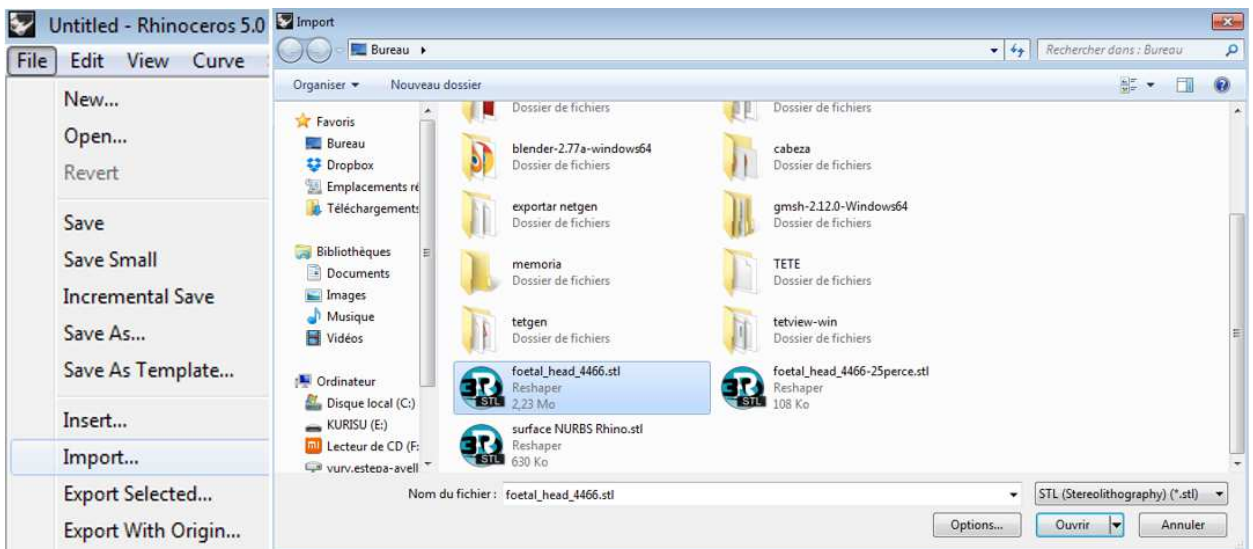


Figure 16 – Importer des fichiers en utilisant Rhinoceros 2015.

avec les extensions *.stl*, *.obj* et *.wrl*, entre autres options.

On a utilisé le fichier appelé *foetal – head – 4466.stl* pour montrer la procédure suivie. Nous chargeons le fichier et la fenêtre des paramètres d'importation apparaît, comme le montre la figure 17.

Il permet de changer les options suivantes :

- **Weld Angle** : Cette option permet de choisir l'angle de soudure du maillage.

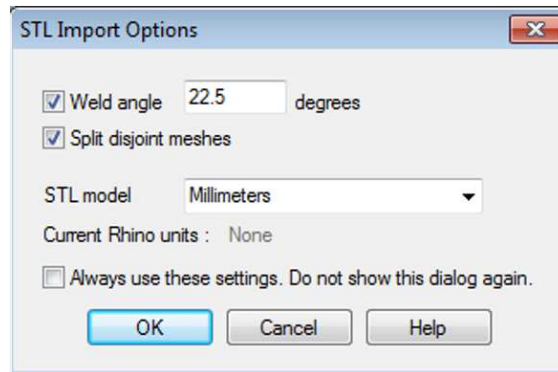


Figure 17 – Options d’importation fichiers avec formats .stl dans le logiciel Rhino.

- **Split disjoint meshes** : S’il y a des problèmes dans le maillage en choisissant cette préférence, le logiciel détermine si les pièces disjointes d’une maille seront automatiquement partagées sur l’importation.
- **STL Model** : Si le fichier .stl a des unités, Rhino permet de visualiser et choisir les unités appropriées.
- **Always use these settings** : Et finalement, on a le choix de sauvegarder les configurations pour les importations prochaines.

On a utilisé les préférences montrées dans la figure 17, et Rhinoceros nous offre une interface avec 4 vues possibles comme la figure 18 illustre et selon l’aide de Rhinoceros[11].

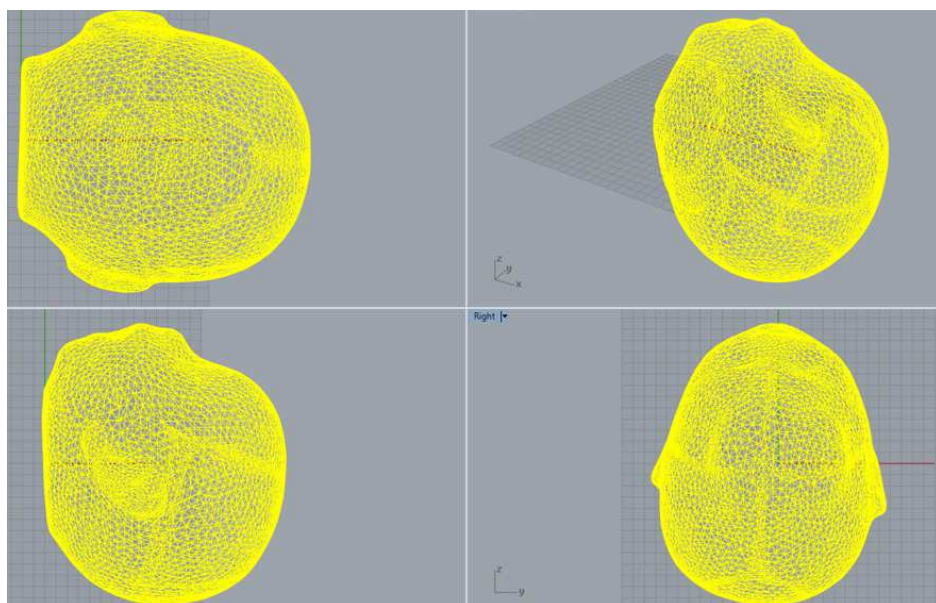


Figure 18 – Espace de travail dans le logiciel Rhinoceros 2015.

2. Résolution : Nombre de polygones et triangles

Comme dans le cas antérieur avec 3DS Max, on a beaucoup de polygones et nous avons besoin de diminuer le nombre de points afin de travailler facilement avec le maillage. Rhino permet de connaître les propriétés de l'objet. Mais, on doit analyser chaque maillage, afin de décider si c'est nécessaire faire la réduction ou pas.

Pour observer les caractéristiques principales, on sélectionne le Menu *View* et après on choisit l'option *Display Options*. Ensuite, la fenêtre avec les options de visibilité sont montrées comme le montre la figure 19 ci-dessous.

L'utilisateur peut choisir les options qu'il veut voir, dans notre cas, on a sélectionné

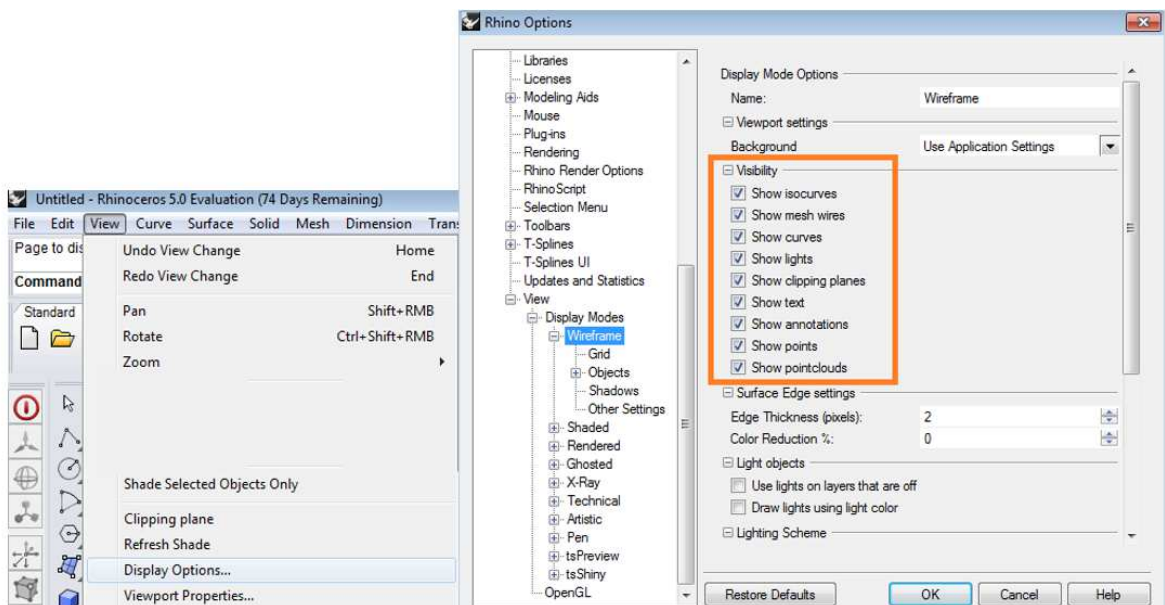


Figure 19 – Préférences de visibilité Rhinoceros 2015

toutes les possibilités qui sont soulignées en utilisant le rectangle orange de la figure 19.

Puis, on pourra voir toutes les caractéristiques, dans le menu *Propriétés* qui se trouve à droite dans la zone de travail de Rhino. On choisit *Details* et immédiatement une fenêtre avec les descriptions de l'objet apparaît comme illustré dans la figure 20.

Dans ce tableau, on peut voir :

- Le numéro de vertex : 4596
- Le numéro de polygones : 8928

En utilisant ces données, on va commencer la réduction de nombre de polygones, triangles et vertex. En premier lieu, on va sélectionner le menu *Mesh*. Ensuite, l'option *Mesh Edit Tools -> Collapse -> Reduce Vertex Count*. De manière optionnel, on peut sélectionner le menu *Mesh*. Ensuite, l'option *Mesh Edit Tools -> Reduce Mesh*.

Il s'agit d'un autre chemin qui permet de réduire le maillage. Après, on peut voir la fenêtre de la figure 21.

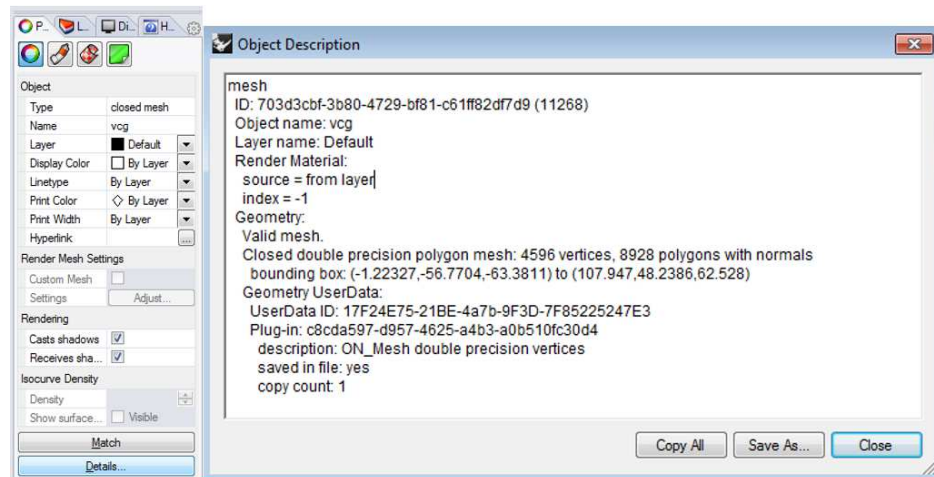


Figure 20 – Propriétés du maillage dans le logiciel Rhinoceros 2015

La réduction peut être faite en changeant le nombre de polygones, par pourcentage

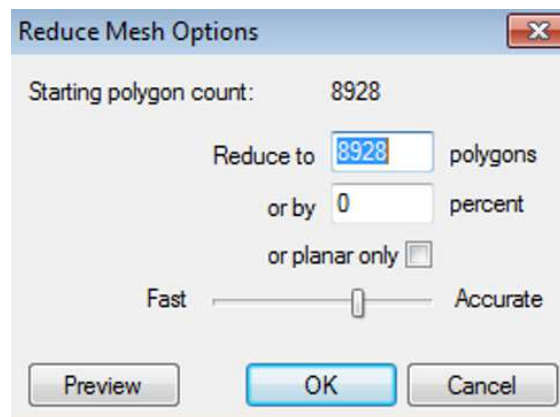


Figure 21 – Options de réductions de polygones dans le logiciel Rhinoceros 2015

ou uniquement restreindre de la réduction aux polygones plans.

De même, on peut choisir le type de réduction que l'on veut : une réduction rapide ou une réduction précise.

Les différences sont les suivantes :

- *Rapide* : La réduction est réalisée plus aléatoirement, avec moins de calcul, ce qui permet d'être plus rapide.
- *Précise* : La réduction produit le moins possible d'artefacts visibles. Ceci implique plus de calcul et donc une vitesse plus lente.

On va commencer la réduction de 8928 polygones jusqu'à 4464 polygones. De cette façon, on obtiendra une réduction comme l'illustre la figure 22.

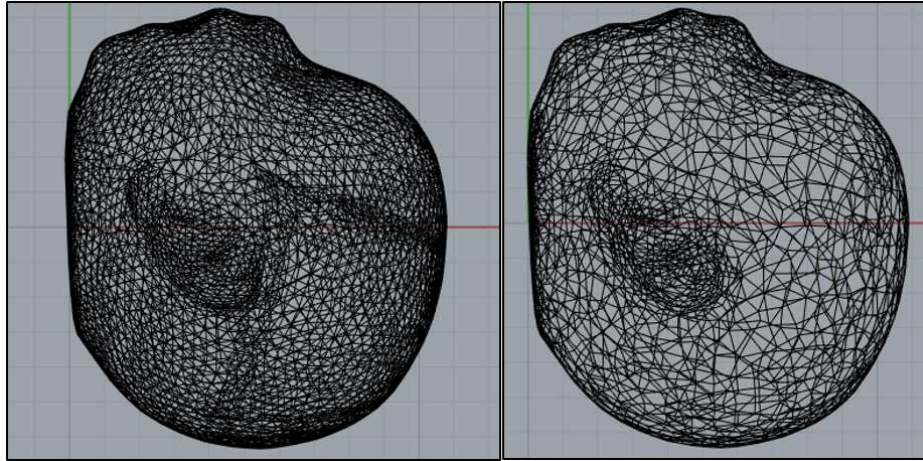


Figure 22 – Réductions de polygones effectués à la tête foetale sur le logiciel Rhinoceros 2015.

2.2.2 Méthode en utilisant Rhinoresurf

1. *Conversion Maillage à surfaces de manière automatique :*

En premier lieu, pour faire la conversion à Nurbs, c'est nécessaire que nous téléchargeons le plugin s'appelle *Rhino Resurf* depuis sa page officielle. Il y a un trial avec une durée de 20 jours.

Le premier pas est choisir le maillage de la tête foetale, dans notre cas. Après, nous allons chercher le Menu Rhino Resurf qui va apparaître depuis l'installation du plugin. Du coup, on va sélectionner le menu *Rhino Resurf -> Automatic surfaces generation from mesh*. Puis, Rhinoceros montrera la figure 23.

De cette manière, le logiciel nous donnera la fenêtre de paramétrage pour la génération d'un réseau de courbes et pour la surface (Figure 23). Ensuite, il faut que nous choisissons l'option *Curve Network* et de cette façon, on va obtenir un réseau des courbes comme le montre la figure 24 ci-dessous.

Le pas suivant est de vérifier chaque courbe, afin de s'assurer qu'il n'y aura aucun problème dans la génération du surface Nurbs. Par exemple, l'idée est de trouver les courbes adoucies pour faciliter la conversion. S'il y a des problèmes, on doit utiliser l'outil *Smooth Curve*. On va montrer le processus que on a suivi pour améliorer la courbe de la figure 25.

Pour corriger ces erreurs, on sélectionnera la courbe qui a le problème et après on utilisera l'option *Smooth Curve* du menu *Automatic surface generation* qui montre la figure 23. On va répéter le processus avec toutes les courbes qui ont le même problème.

S'il y a besoin de construire quelques lignes, on peut utiliser la commande *PolylineOnmesh*. Cet outil est facile d'emploi et il exige seulement une déclaration du point

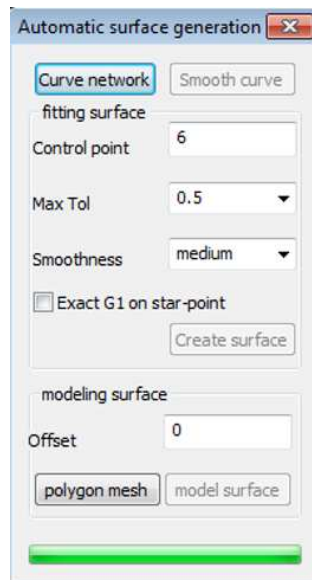


Figure 23 – Fenêtre de génération automatique des courbes et surfaces dans le logiciel Rhinoceros 2015.

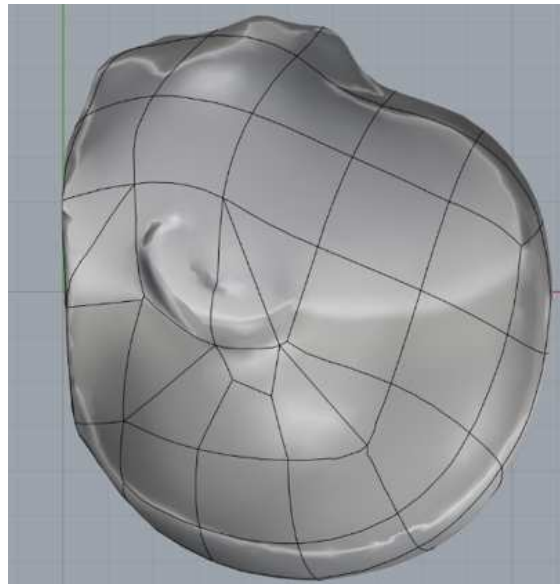


Figure 24 – Réseau des courbes dans le maillage de la tête foetale dans le logiciel Rhinoceros 2015.

de début et de la fin de polyline.

Quand le réseau des courbes est adéquat, on va créer la surface en suivant ce processus. Il faut que nous sélectionnions le nombre de point de contrôle (*Control Point*) dans la fenêtre de la figure 26 et après, on va sélectionner l'option *Create Surface*.

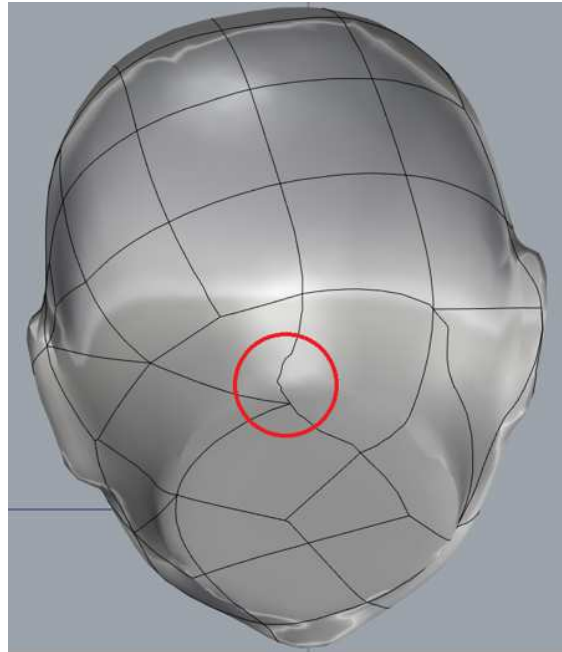


Figure 25 – Problème dans une courbe du maillage de la tête foetale dans le logiciel Rhinoceros 2015.

Quand, nous avons terminé la création de la surface, on peut changer la visibilité des

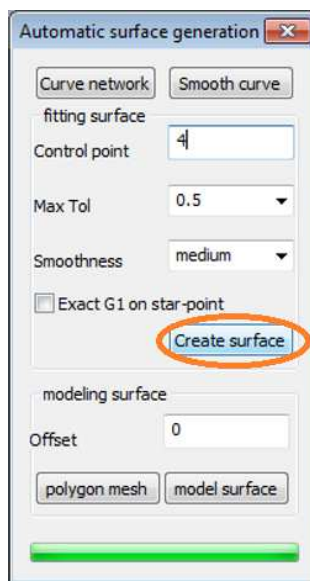


Figure 26 – Génération automatique des surfaces dans le logiciel Rhinoceros 2015.

autres couches de l'objet, dans le menu *Layers* qui se trouve dans la partie droite de la zone de travail de Rhinoceros. De cette façon, on peut voir de manière précise la surface, comme le montre la figure 27.

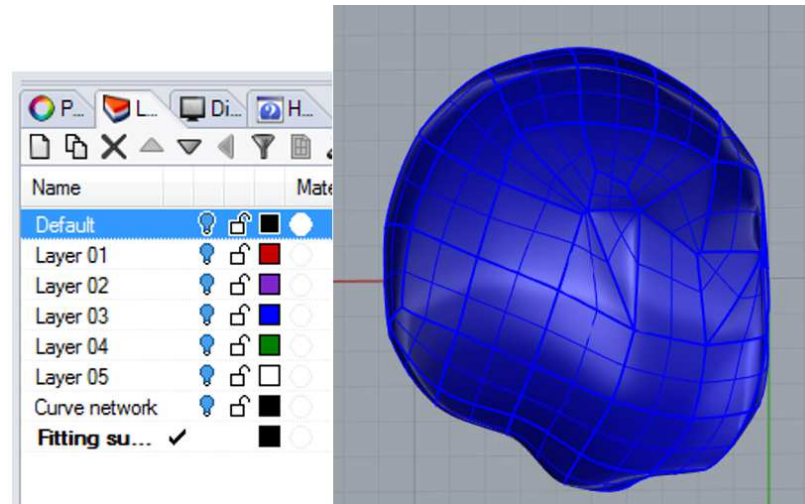


Figure 27 – Affichage des surfaces dans le logiciel Rhinoceros 2015.

De même, nous pouvons manipuler les points de contrôle en utilisant la commande *PointsOn*, le bouton F10 ou sélectionnant le Menu *Edit-> Control Points -> Control Point On* et finalement, on obtiendra les points de contrôle de notre surface comme illustré la figure 28 ci-dessous.

Avec cet outil, nous pouvons faire des changements sur notre surface facilement. Pour

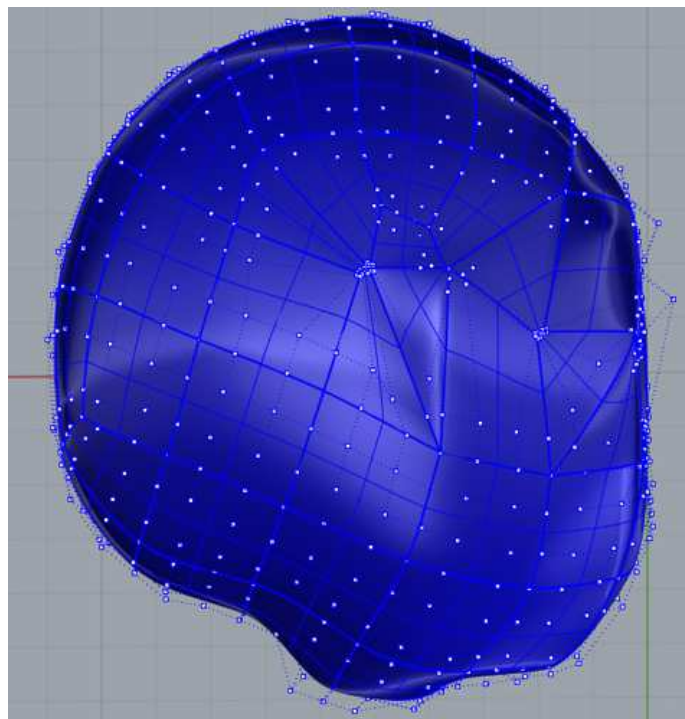


Figure 28 – Affichage des points de contrôle de la surface dans le logiciel Rhinoceros 2015.

ne pas montrer les points de contrôle, il suffit de taper F11.

2. *Exportation du maillage à format .stl*

Par la suite, on fera l'exportation du fichiers, en utilisant le format *.stl* . On doit générer un maillage de polygones qui peut être exporté en utilisant le format *.stl*

Le premier pas est de construire le maillage de polygones en utilisant l'outil *Polygon Mesh*. Rhinoceros va créer un maillage en prenant en compte le *Fitting Surface*, que nous avons créé dans la section antérieure (Figure 29). Ensuite, on va au menu *File*

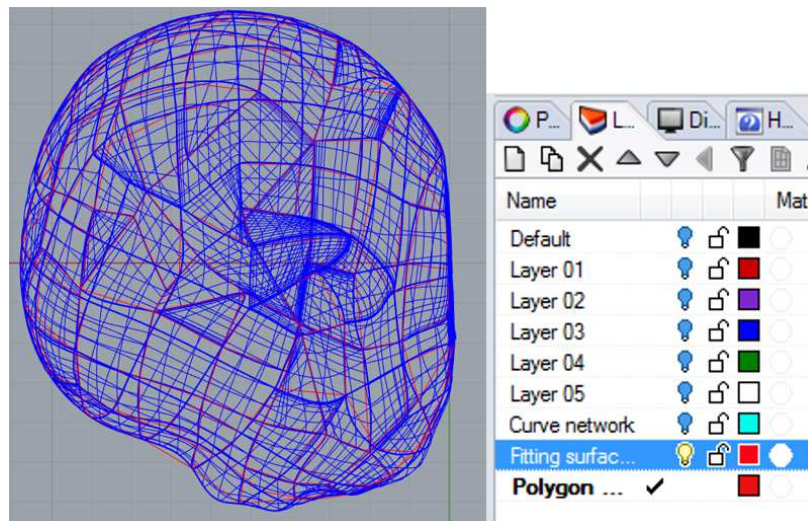


Figure 29 – Affichage de Polygon Mesh dans le logiciel Rhinoceros 2015.

-> *Export selected*. Après, on choisit le maillage (ligne rouge dans ce cas) dans la zone de travail. Rhinoceros nous montrera les fenêtres suivantes (Figure 30), ces options nous permettent d'exporter le maillage avec les paramètres dont nous avons besoin.

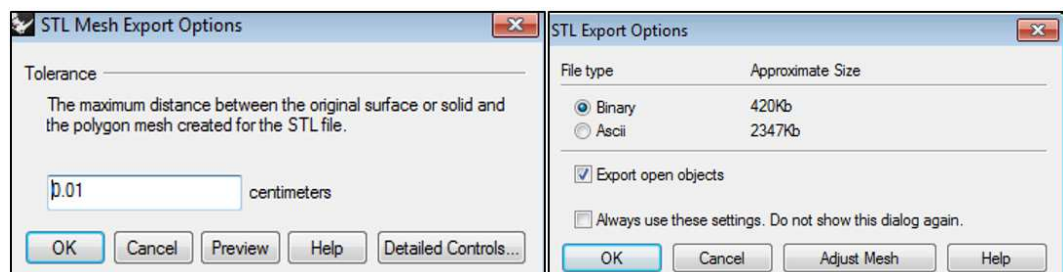


Figure 30 – Paramétrage de l'exportation en utilisant le format *.stl* dans le logiciel Rhinoceros 2015.

2.2.3 Méthode en utilisant Cage Edit

Il y a d'autres chemins pour obtenir un maillage manipulable rapidement. Il s'agit d'utiliser le maillage original *foetal – head – 4466* dans le format *.stl*. On peut suivre les mêmes instructions qu'ont été montrées dans la section *Aménagement du maillage* (1.2.1).

D'abord, on sélectionne le menu *Transform -> Cage Editing -> Cage Edit* comme illustré dans la figure 31. Rhino nous demande de sélectionner le maillage à traiter (*Select Captive*

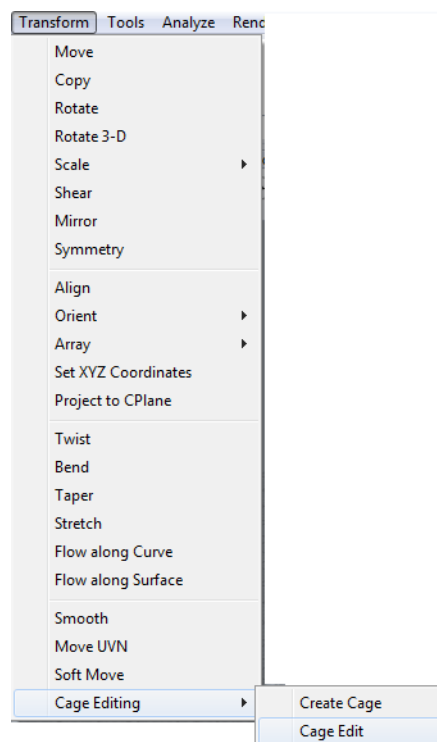
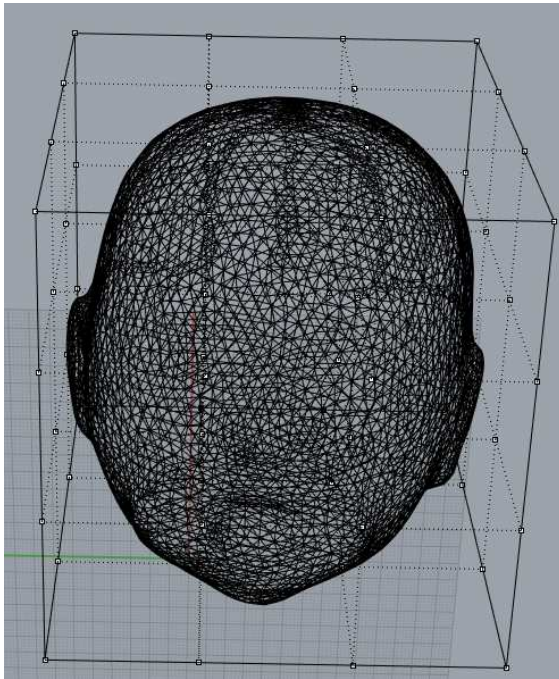
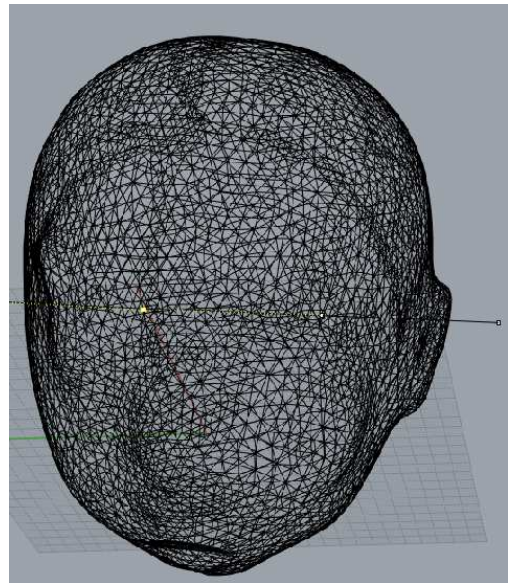


Figure 31 – Menu Transform Rhinoceros 5.0

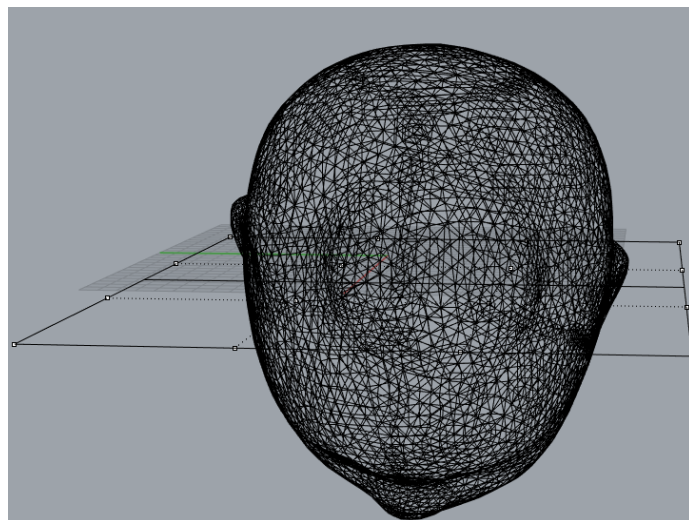
Objets). Après, on doit choisir un objet pour faire le contrôle entre les options suivantes *Boundingbox, Line, Rectangle, Box*. Dans la figure 32, on observe les différentes options appliquées sur la tête foetale.



(a) Cage Boundingbox



(b) Cage Line



(c) Cage rectangle

Figure 32 – Options Cage Edit

Chaque option permet d'établir des paramètres de création. Il y a des actions par défaut, mais il est possible de changer selon les besoins. De même, on peut modifier les options *Deformation* et *PreserveStructure* qui nous fournissent les possibilités suivantes :

- Déformation :
 - *Précise* : La mise à jour est lente, mais les surfaces peuvent être plus denses lorsque les objets déformés sont optimisés.
 - *Rapide* : Cette option génère une surface avec moins de points de contrôle.
- Conserver la structure :

Elle permet de conserver le maillage ou pas. Par défaut, cet outil montre l'option *No*. De cette façon, on peut faire des modifications et des déformations précises.

La structure de la tête foetale que nous avons, nous donne la possibilité de travailler avec une cage de type *boîte de contour* (*Bounding box*). Désormais, on montrera les paramètres que nous avons en choisissant cette option. En premier lieu, on fixera le système de coordonnées. Il y a trois options :

- *CPlane*
- *World*
- *3Point*

Le système de coordonnées, le plus utilisé et celui qui se trouve par défaut est *CPlane*. On va sélectionner ce système.

Après, on doit choisir la quantité de points et le degré de la cage dans chacune des trois directions. Rhino montre par défaut les paramètres suivants :

Cage points :

- *XPointCount=4*
- *YPointCount=4*
- *ZPointCount=4*
- *XDegree=3*
- *YDegree=3*
- *ZDegree=3*

Cependant, vous pouvez modifier la quantité de points selon vos besoins. Dans ce cas, il y a seize points de contrôle par chaque côté. Et finalement, on doit choisir la région que nous voudrions modifier. Ici, Rhino nous montrera les options suivantes :

- *Global*
- *Local*
- *Other*

Les résultats finaux se montrent dans la figure 33-a. Ensuite, on peut faire des modifications en utilisant les points de la cage, comment le montrent la figure 33-b.

Quand on aura fini les modifications du maillage en utilisant la cage, on va suivre les indications qui ont été montrées dans la Section 1.2.2.1 *Exportation du maillage à format .stl*.

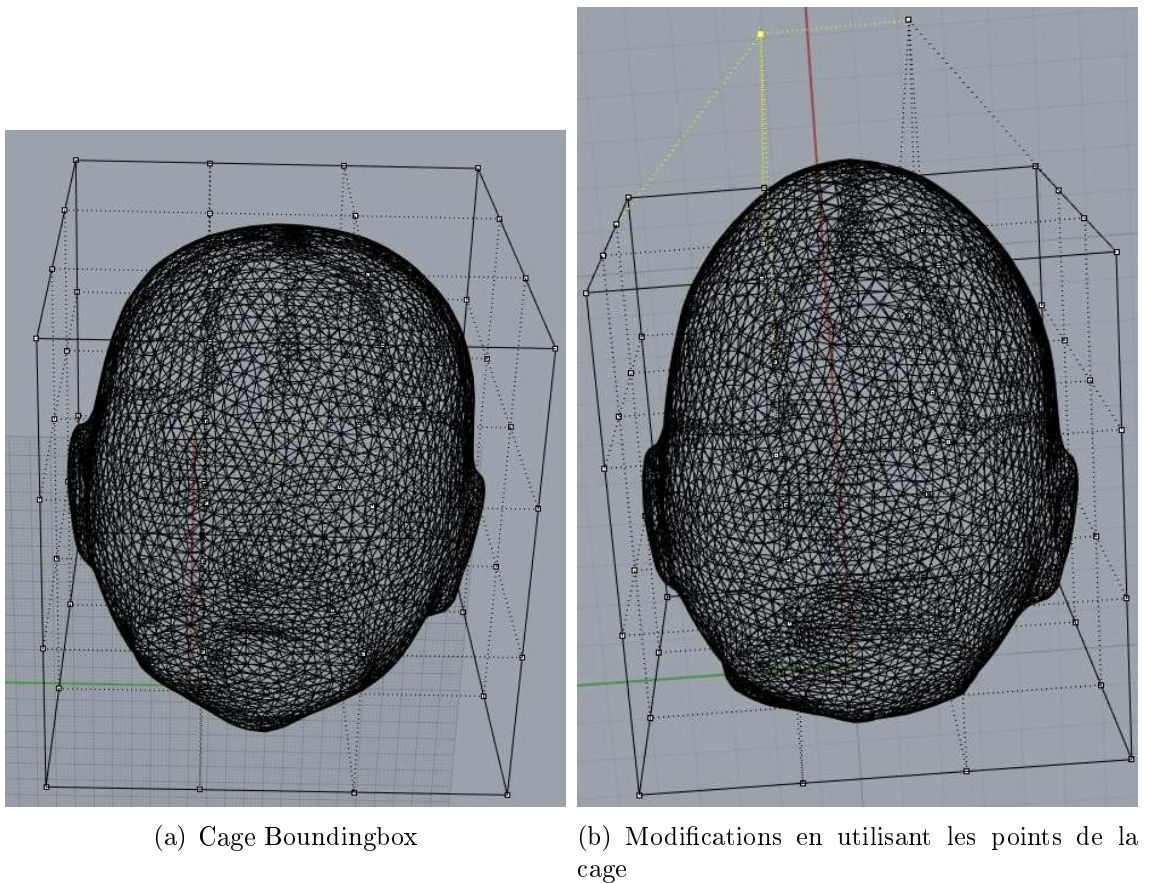


Figure 33 – Outil Cage Edit de logiciel Rhinoceros

2.2.4 Difficultés dans la conversion en utilisant Rhinoceros

La conversion du maillage en utilisant Rhino offre beaucoup des options afin d'obtenir une surface. Cependant, il y a le risque d'obtenir des surfaces qui ont les normales inversées. Ce problème provoque des soucis dans les traitements suivants, par exemple, dans la création de tétraèdres. C'est toujours nécessaire que nous changions les orientations des normales en utilisant un logiciel comme Meshlab. Nous travaillerons avec cet outil dans le chapitre suivant.

3 Génération des tétraèdres à partir du maillage

Après de ce processus, on peut travailler avec le maillage de polygones généré afin de construire des tétraèdres.

3.1 Meshlab

Le logiciel Meshlab nous aidera avec l'aménagement du maillage après le traitement avec le logiciel Rhinoceros. Meshlab est disponible sur les systèmes d'exploitation Windows, Linux et Mac. [12]

3.1.1 Aménagement du maillage

Ici, on va utiliser le fichier .stl de Rhinoceros. Nous allons faire un nettoyage du fichier .stl en utilisant le logiciel *Meshlab*. L'idée est de garantir que toutes les triangles sont fermés et qu'il n'y aura pas de surfaces ouvertes. Cette procédure nous aidera après dans la génération des tétraèdres. Ici, on va sélectionner le menu *File -> Import Mesh*, on cherchera le fichier à traiter et on appuiera sur *Ouvrir*.

Meshlab nous montrera la fenêtre suivante(Figure 34). Dans l'espace de travail, de manière

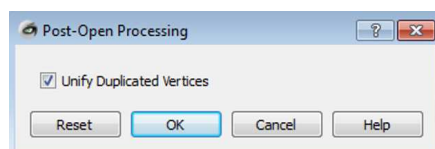


Figure 34 – Paramétrage pour l'importation des fichiers .stl dans le logiciel Meshlab.

immédiate, notre maillage apparaîtra comme le montre la figure 35.

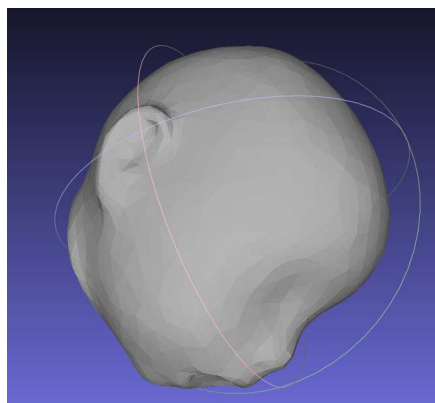


Figure 35 – Tete foetale dans le logiciel Meshlab.

3.1.2 Résolution des problèmes

Il y aura quelques fois des erreurs comme le montre la figure 36. On peut voir qu'il y a un souci avec la majorité des triangles. Les triangles de couleur noire ont les normales de leurs faces à l'intérieur et les triangles de couleur grise ont les normales de leurs faces à l'extérieur.

Pour résoudre cet inconvénient, on peut utiliser l'option de correction des normales de

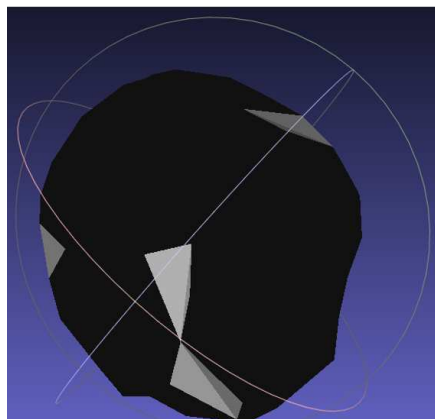


Figure 36 – Erreurs d'orientation des normales du maillage.

Meshlab. Nous la trouverons dans le menu *Filters->Normals, Curvatures and Orientations ->Invert Faces Orientations* (Figure 37) .

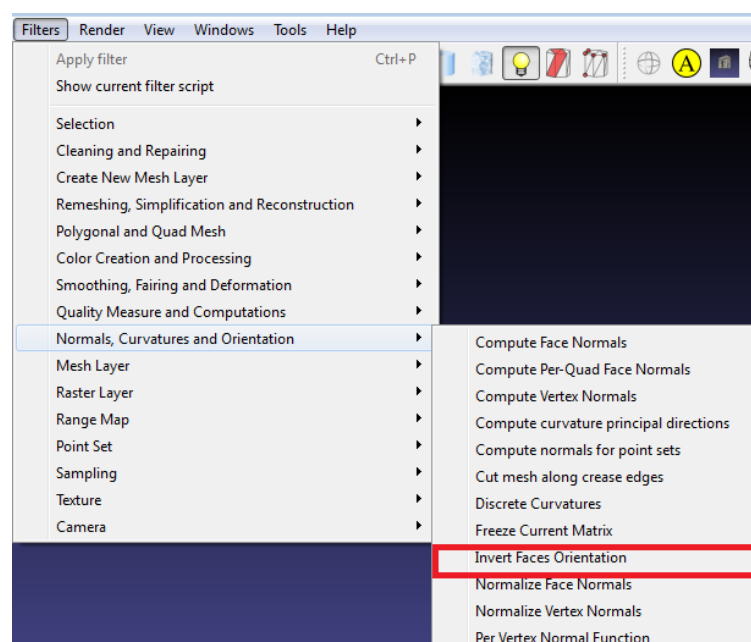


Figure 37 – Changement des orientations des normales dans le logiciel Meshlab

3.1.3 Exportation du maillage à format .stl

D'abord, on fera une exportation en utilisant le format .stl. Mais dans ce cas, on va garder sans le codage binaire. On cherche le menu *File -> Export Mesh as*. Et après, on suivra les paramètres qui se montrent dans la fenêtre suivante (figure 38).

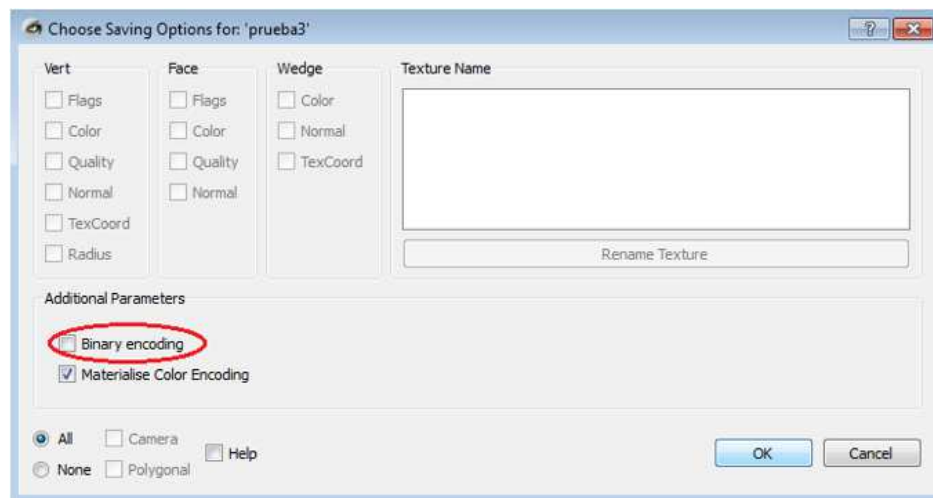


Figure 38 – Options pour faire l'exportation en utilisant le format .stl dans le logiciel Meshlab.

3.2 Netgen 5.0

3.2.1 Conditionnement du maillage, format .stl :

En premier lieu, on a besoin d'autre logiciel qui nous permet de faire la construction de tétraèdres. Pour cette raison, on va travailler en utilisant *Netgen 5.0*. Ce logiciel est disponible sur les systèmes d'exploitation Windows, Linux et Mac. Netgen nous donne l'option d'ouvrir des fichiers avec les suivants formats *.igs*, *.iges*, *.stl*, *.stlb*, *.geo* et d'autres. [13] Quand, on ouvrira Netgen, on ira au Menu *File-> Load Geometry*. Ensuite, on va choisir le fichier *.stl*.(Figure 39)

3.2.2 Résolution des problèmes

Il y aura quelques fois des erreurs comme le montre la figure 36.

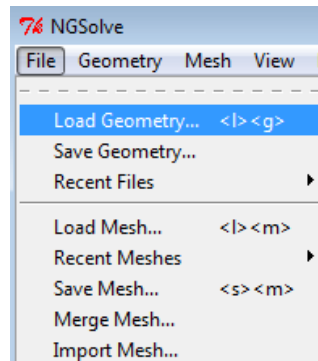


Figure 39 – Maillage de la tête foetale dans le logiciel Netgen 5.0

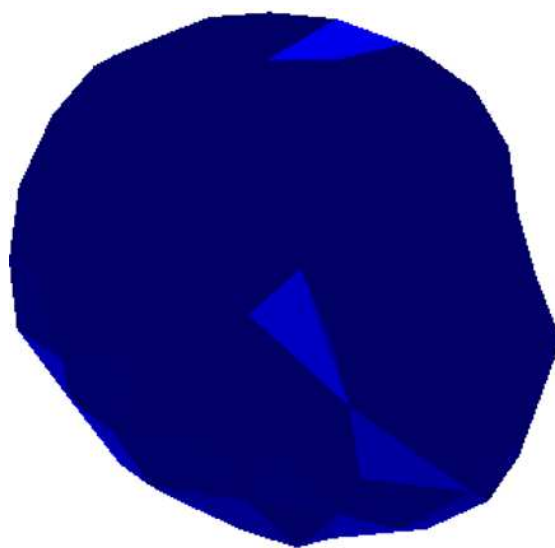


Figure 40 – Maillage de la tête foetale dans le logiciel Netgen 5.0

La géométrie, de la figure 40, a les caractéristiques suivantes :

- Les triangles brillants ont les faces à l'extérieur.
- Les triangles opaques ont leurs faces à l'intérieur.

Le deuxième groupe provoquera des problèmes dans la génération des tétraèdres. Il faut qu'on change l'orientation des triangles. (Si nous ne l'avons pas fait en utilisant Meshlab). On utilisera l'outil *STL INFO* de Netgen pour connaître l'état initial du maillage. De même, on peut voir la sortie de la console de Netgen comme illustré dans la figure 41. Clairement,

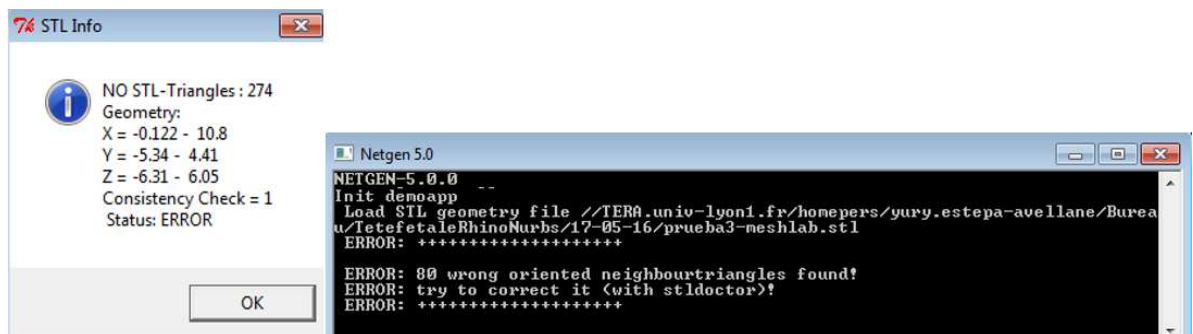


Figure 41 – STL INFO et l'info dans la console dans le logiciel Netgen 5.0

il y a une erreur dans notre maillage. Pour corriger les triangles, nous utiliserons l'outil *STL DOCTOR* qui vient avec Netgen. On va commencer en choisissant la possibilité de sélectionner des triangles en faisant double click dans le menu *General* de *STL DOCTOR*. Ensuite, on sélectionnera les triangles à changer un pour un dans l'espace du travail et puis, on va choisir le menu *Edit Topology*.

Ce menu permet de voir les inconsistances par rapport aux triangles du maillage (option *mark inconsistent triangles*) et aussi il permet de modifier l'orientation des triangles. On utilisera l'option *Orient after selected trig* (Figure 42).

Dans la suite, Netgen nous montrera la tête foetale, comme le montre la figure ci-dessous. De même, on peut regarder l'état du maillage, afin de constater que le maillage est bon (Figure 43). On peut fermer la fenêtre de *STL DOCTOR*.

S'il y a d'autres soucis, il y a d'autres outils de *STL DOCTOR* qui permettent de corriger les normales. (Figure 44)

3.2.3 Génération des tétraèdres à partir du maillage :

Une fois qu'on aura le maillage sans aucun souci. On va suivre la procédure suivante :

1. *Génération du maillage* :
On doit sélectionner le menu *Mesh-> Generate Mesh*.
2. *Vérification de qualité* :
On doit choisir dans le menu *Mesh-> Mesh Quality*.

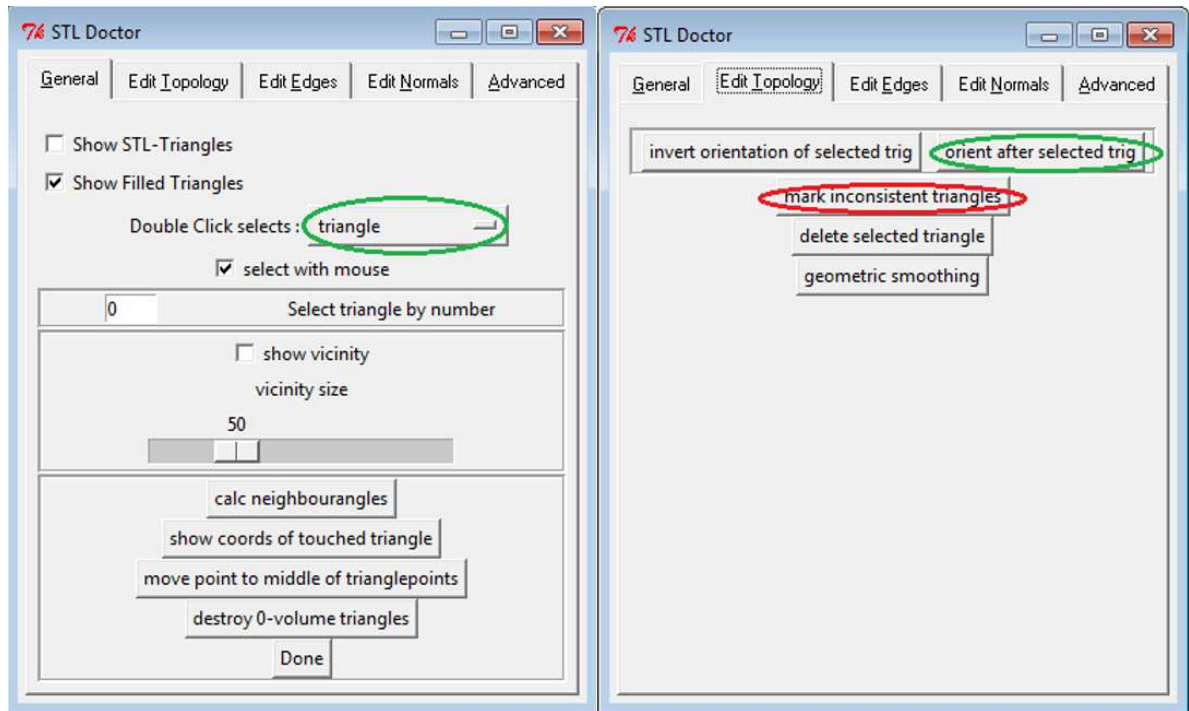


Figure 42 – STL DOCTOR dans le logiciel Netgen 5.0

3. *Analyse de Géométrie :*

On doit choisir dans le menu *Mesh-> Analyse Geometry*.

4. *Generation Surface du maillage :*

Puis, on choisira le menu *Mesh-> Mesh Surface*

5. *Optimisation de Surface :*

Après, c'est nécessaire de faire une optimisation de surface en accédant depuis le menu *Mesh-> Optimize Surface*.

6. *Generation Volume :*

Finalement, on va construire le volume pour notre maillage en utilisant le *Mesh-> Mesh Volume*.

7. *Optimisation Volume :*

Pour bien finir avec notre volume, on fera la dernière optimisation a travers du *Mesh-> Optimize Volume*.

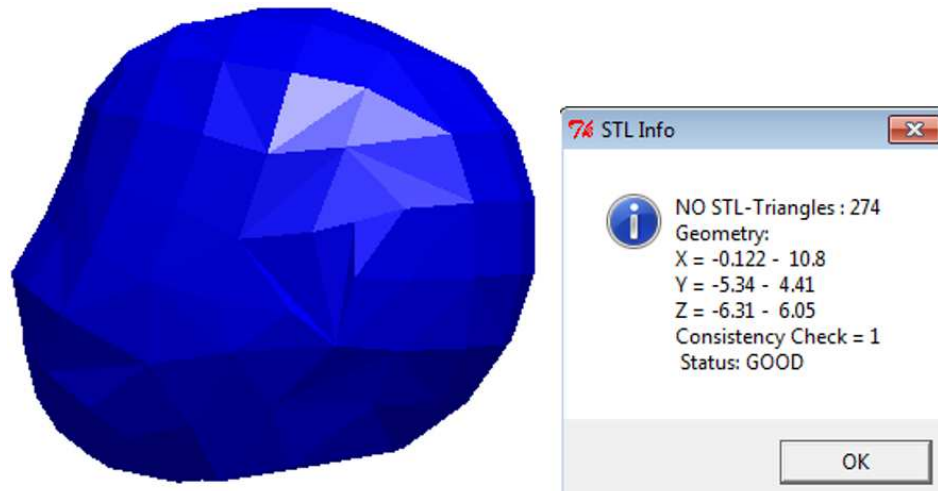


Figure 43 – Maillage de la tête foetale après les corrections en utilisant l’outil STL DOCTOR dans Netgen 5.0

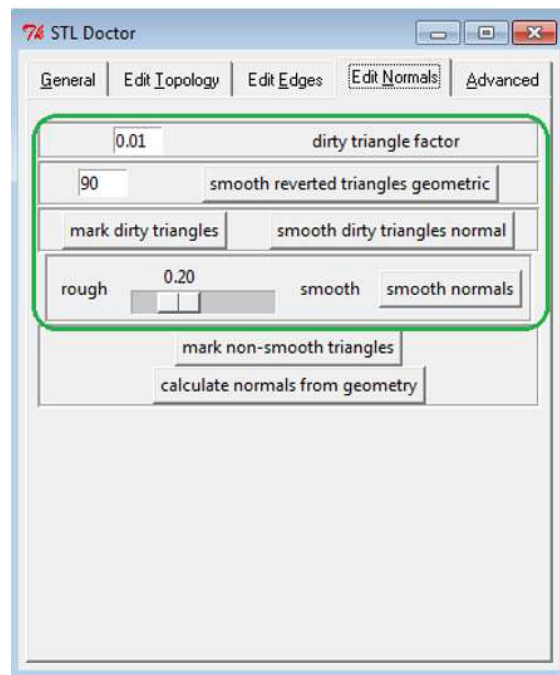
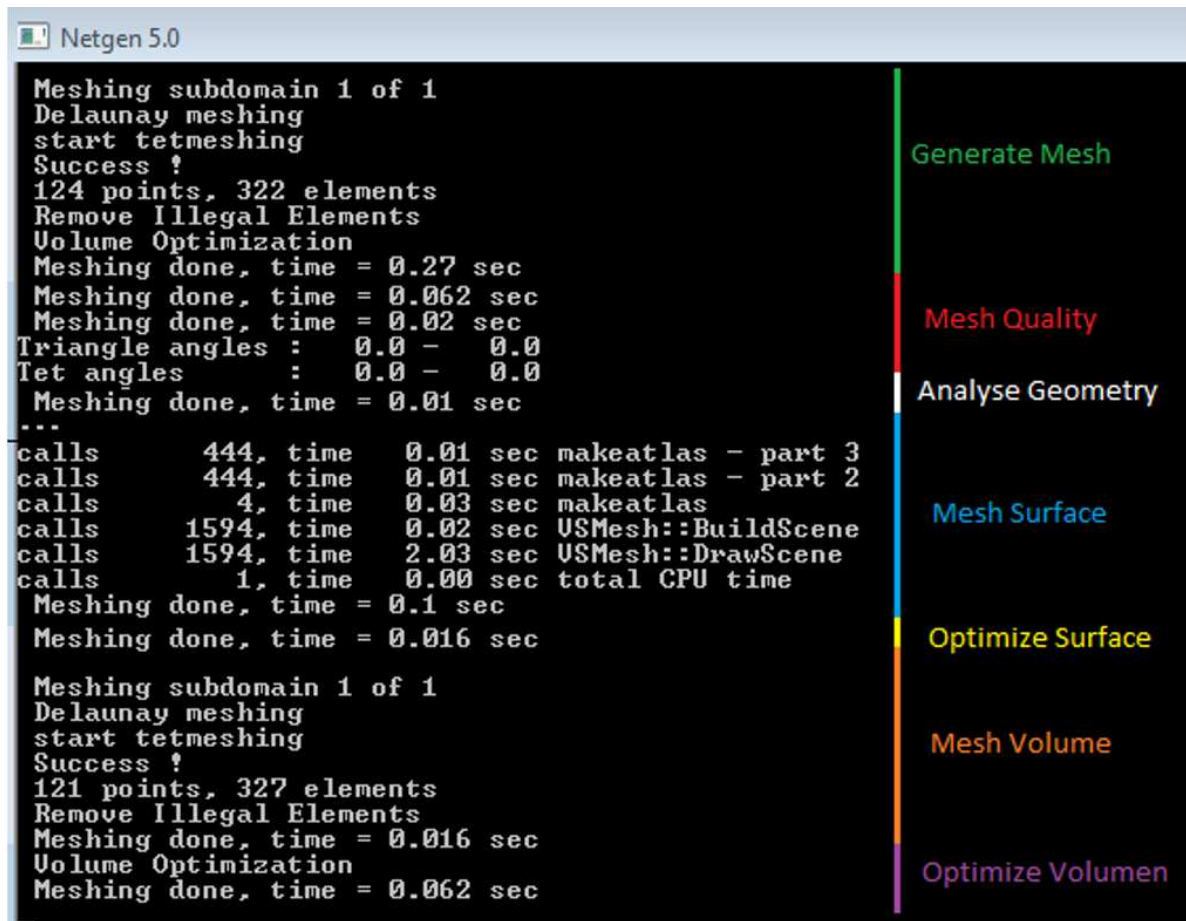


Figure 44 – D’autres outils STL DOCTOR dans le logiciel Netgen 5.0

Dans la figure 45, vous pouvez voir la sortie de la console pour chaque opération faite.



```

Netgen 5.0
Meshing subdomain 1 of 1
Delaunay meshing
start tetmeshing
Success !
124 points, 322 elements
Remove Illegal Elements
Volume Optimization
Meshing done, time = 0.27 sec
Meshing done, time = 0.062 sec
Meshing done, time = 0.02 sec
Triangle angles : 0.0 - 0.0
Tet angles : 0.0 - 0.0
Meshing done, time = 0.01 sec
...
calls 444, time 0.01 sec makeatlas - part 3
calls 444, time 0.01 sec makeatlas - part 2
calls 4, time 0.03 sec makeatlas
calls 1594, time 0.02 sec USMesh::BuildScene
calls 1594, time 2.03 sec USMesh::DrawScene
calls 1, time 0.00 sec total CPU time
Meshing done, time = 0.1 sec
Meshing done, time = 0.016 sec

Meshing subdomain 1 of 1
Delaunay meshing
start tetmeshing
Success !
121 points, 327 elements
Remove Illegal Elements
Meshing done, time = 0.016 sec
Volume Optimization
Meshing done, time = 0.062 sec

```

Figure 45 – Sortie de la console Netgen pour toute la procedure de génération des tétraèdres

3.2.4 Exporter du maillage en utilisant le format gmsh2

Après avoir suivi les pas antérieurs, on doit sélectionner le format correct pour faire l'exportation de notre volumen tétraédrique.

On doit chercher dans le *Menu File->Export file Type->Gmsh2 Format.*, comme le montre la figure 46. Ensuite, vous pouvez choisir l'option *export Mesh* de *Menu File*, et vous devez sélectionner un nom pour votre maillage et le processus sera terminé. Dans la figure 47, on peut voir la tête foetale après la génération des tétraèdres.

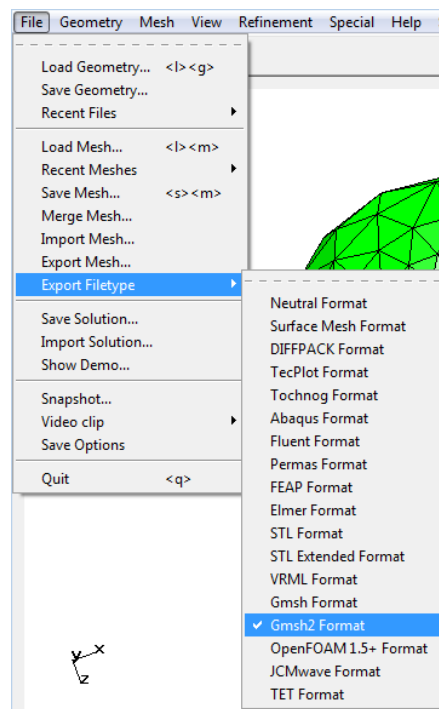


Figure 46 – Exporter en utilisant le format gmsh2 sur Netgen

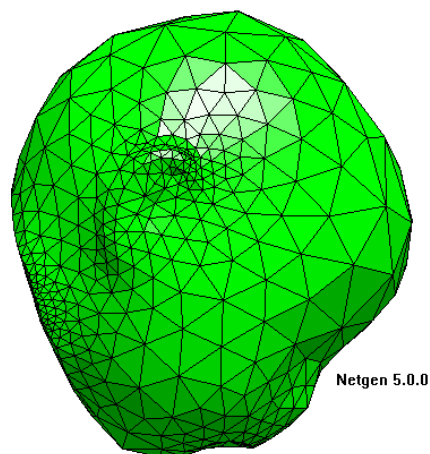


Figure 47 – Tête foetale après de la génération des tétraèdres

3.2.5 Caractéristiques du format gmsh2

Le format .gmsh2 de notre tête foetale a les champs suivants :

— *Information générale du maillage :*

```
$MeshFormat
2.2 0 8 -> Version - type de fichier - taille du fichier
$EndMeshFormat
```

— *Information spécifique selon le cas (Nodes, elements, Region names, periodic relations and post-processing datasets)*

```
$Nodes
5                cinq noeuds dans le maillage:
1 0.0 0.0 0.0    noeud #1: coordinates (0.0, 0.0, 0.0)
2 1.0 0.0 3.0    noeud #2: coordinates (1.0, 0.0, 3.0)
3 1.0 4.0 7.0    etc.
4 0.0 1.0 0.0
5 3.0 0.0 1.0
$EndNodes
$Elements
2                deux éléments
1 4 2 99 2 1 2 3 4  Numéro d'élément - type d'élément
                  - nombre de tags - list de noeuds
2 4 2 101 2 21 8 16 3  Élément #2: type 4 (tetraedron), physical 101,
                  elementary 2, nodes 21 8 16 3
$EndElements
```

Les éléments peuvent être de différents types. Par exemple :

- *Type 1* : Noeud type ligne
- *Type 2* : Noeud type triangle
- *Type 3* : Noeud type quadrangle
- *Type 4* : Noeud type tetradron

4 Detection des pressions exercées sur la tête foetale

Pour certains accouchements, il faut que les médecins utilisent des outils d'extraction instrumental qui permettent de saisir la tête foetale pendant la naissance.

Il y a deux outils les plus connus : les forceps et la ventouse.

Le forceps est constitué par deux branches qui sont introduites de manière séparée. Ensuite, quand les branches sont bien placées, le responsable de la manoeuvre peut les attacher et les articuler ensemble.

Dans la figure 48, on peut voir les pas à tenir en compte pour la mise en place des branches du forceps.

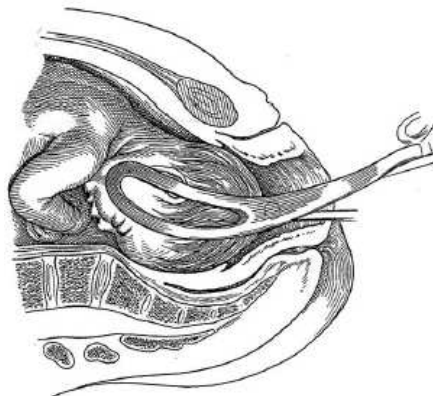


Figure 48 – Mise en place du Forceps.[14]

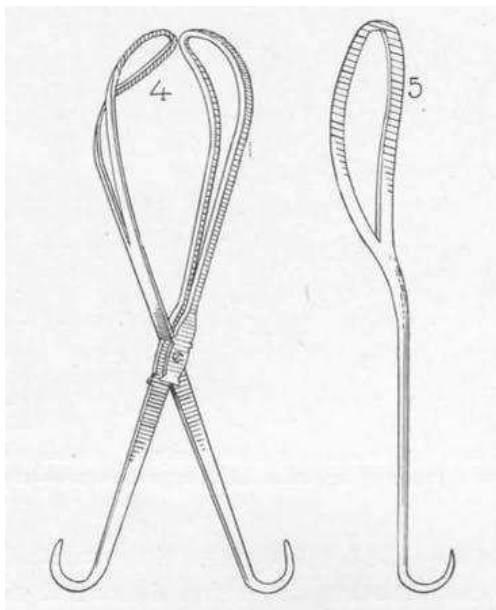
Cependant, il y a eu quelques complications en utilisant le forceps, par exemple les fistules et les déchirures vaginales et l'excès de pression sur la tête foetale. Notre application a le but de trouver et calculer les forces qui sont appliquées en utilisant le forceps sur la tête foetale.

4.1 Forceps

Le forceps est un outil utilisé depuis longtemps. On considère que l'inventeur est Monsieur Peter Chamberlain, parce qu'il a été la première personne à utiliser le forceps en séparant les branches de la pince([15]). De même, Monsieur Levret a modifiée le forceps de Chamberlain ([16]) et il a crée la courbure pelvienne dans les cuillères de forceps afin de faciliter l'extraction de la tête foetale et suivre le chemin pelvien. Après, Etienne Tarnier a amélioré le forceps de Levret en ajoutant un tracteur qui a permis de mettre en valeur le dernier dessin et de cette manière, les personnels de santé peuvent suivre la trajectoire de la tête foetale. ([16])

À l'époque actuelle, il y a beaucoup des modèles de forceps selon le pays, de l'école de Médecine et les traditions de la région. Il y a plusieurs variations par rapport aux forceps. Il y a des modèles avec les bras plus courts que d'autres et il y a aussi d'autres modèles qui vient avec tracteur.

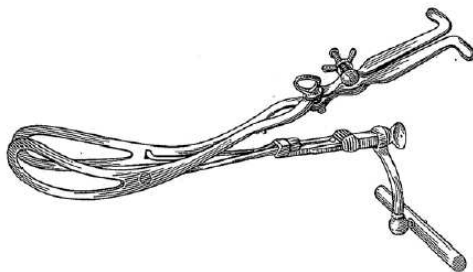
Dans la figure 49, se montrent quelques modèles.



(a) Forceps Chamberlain. [17]



(b) Forceps Levret. [18]



(c) Forceps Tarnier avec tracteur.[14]



(d) Forceps Pajot. [19]

Figure 49 – Modèles du forceps.

4.1.1 Sélection des forceps

Dans les écoles françaises et particulièrement à Lyon, le forceps le plus utilisé est le modèle Levret. Cependant, ce n'est pas facile de trouver ce modèle sur le marché. Pour notre application, on a choisi un forceps type Pajot afin de faire des preuves avec et de fixer les capteurs sur la surface des chaque branche.

4.2 Capteurs de pression

Pour l'exécution des mesures de pression et de la force appliquée sur la tête foetale, on a décidé d'utiliser des capteurs qui permettent obtenir une valeur de pression quand le forceps fait contact avec la tête de nouveau-né.

4.2.1 Sélection des capteurs

Nous avons effectué l'état de l'art par rapport aux capteurs actuels qui nous donnent la possibilité de mesurer cette force. Parmi les capteurs, on peut trouver les catégories suivantes :

- **Capacitives** : Le fonctionnement des capteurs de type capacitif est fondé dans le changement de la valeur de la capacitance entre deux pièces métalliques et un diélectrique à l'intérieur du capteur. Normalement, les tailles des capteurs sont assez petites et pour cette raison, les prix sont élevés.
- **Piézorésistives** : (FSR :Force Sensing Resistor) D'autre côté, il y a les capteurs de type résistifs qui changent la valeur de la résistance par rapport à la pression exercée sur la surface du capteur. Il y a beaucoup de fournisseurs qui offrent différentes options selon la taille, la pression à mesurer, etc. Dans la figure 50, on peut observer différents capteurs *FSR*.



Figure 50 – Capteurs Piézorésistifs : FSR.

- **Matricielles** : Les capteurs de type matricielle sont parfaits pour les applications qui ont besoin de mesurer plusieurs pressions à même temps. Également, les capteurs sont construits en suivant un patron, et cela peut être une contrainte pour notre application, parce que les capteurs de type matriciel ne peuvent pas être coupés ou

modifiés. Dans la figure 51, on trouve un exemple d'un capteur matriciel de Tekscan.

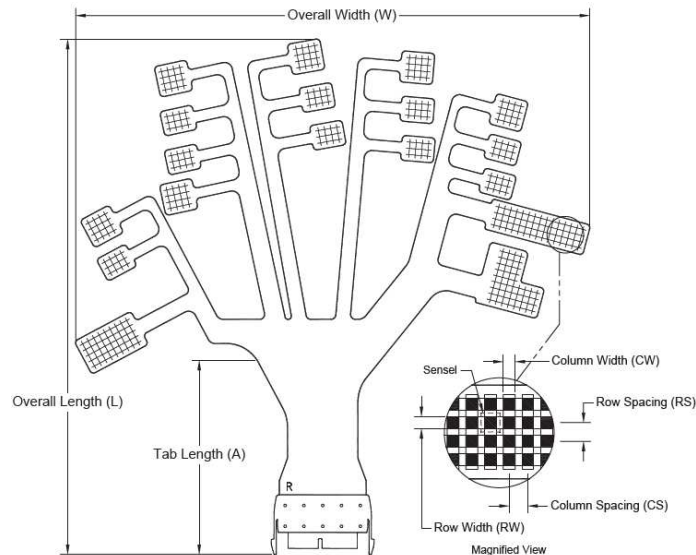


Figure 51 – Capteur matriciel Tekscan

Entre les options, on a choisi les capteurs de type résistif parce qu'ils donnent les meilleures prestations. Ils ne sont pas chers, ils sont flexibles et on peut travailler de manière intuitive avec eux. Nous avons tenu en compte des considérations suivantes :

- Le capteur doit être approprié pour notre application. (Caractéristiques physiques)
- Le prix
- La facilité pour l'exploitation des données.

4.2.2 Caractéristiques des capteurs sélectionnés

Les capteurs CP150 de fournisseurs IEE ont les caractéristiques qui se montrent dans la table 1 :

Taille	12 cm de diamètre
Gamme de pression optimale	0.007 à 7 bars
Résistance	10M Ω à 1k Ω
Pression maximum appliquée :	35 bars
Temps de montée du signal :	1 à 2 ms

Table 1 – Caractéristiques principales du capteur CP150 IEE

On doit rappeler que la pression est la force par unité de surface. Il y a beaucoup d'unités de mesure, selon le champ d'application. Ensuite, vous pouvez voir les plus utilisées.

- **Pascal (Pa)** => $1 \text{ N} / \text{m}^2$
- **Psi** : => $6,894 \times 10^3$
- **At** : => 101325 Pa^2
- **Bar** : => 100.000 Pa
- **mmHg** : =>
- **Torr** :=> $133,322 \text{ Pa}$

Grâce aux conversions antérieures, on connaît la gamme de pression optimale du capteur en tenant compte du système international de mesure (SI) :

0.007 à 7 bars => $700 \text{ Pa (Nm/m}^2)$ à $700000 \text{ Pa (Nm/m}^2)$
 0.007 à 7 bars => 0.07 Nm/cm^2 à 70 Nm/cm^2)

4.2.3 Conditionnement des capteurs

Les capteurs que nous sommes en train d'utiliser sont de type FSR, et c'est la raison pour laquelle on peut considérer que leur fonctionnement sera lié au changement d'une résistance quand on fait pression sur la surface active.

C'est fondamental, alors, de concevoir un circuit qui permet de mesurer les changements de résistance qui varient entre $10 \text{ k}\Omega$ et $1 \text{ M}\Omega$. La considération antérieure est fondée sur la courbe typique de comportement du capteur illustré dans la figure 52.

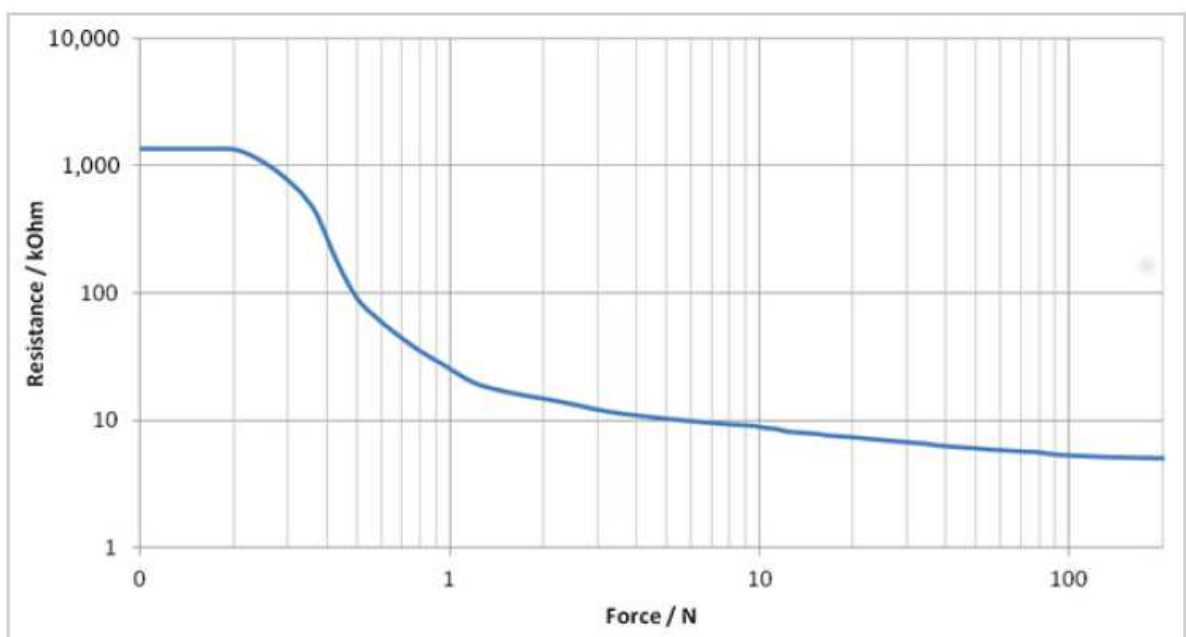
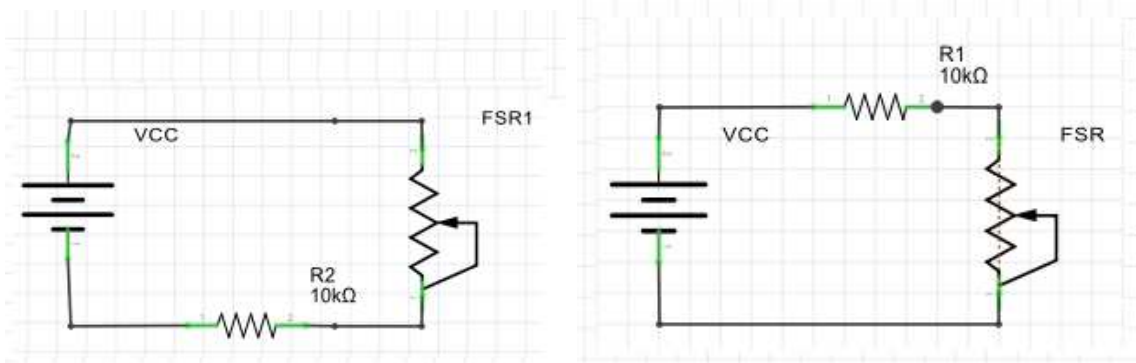


Figure 52 – Courbe typique de comportement Capteur IEE. [20]

Nous avons deux options, qui se montreront ensuite :

1. Méthode Résistance Pull-down ou Pull-Up :

La première méthode pour connaître le changement de résistance de notre capteur est d'utiliser une simple résistance en configuration Pull-down ou pull-up, comme sur la figure 53. L'inconvénient est que si la résistance du capteur change entre $1k\Omega$ et



(a) Configuration de résistance Pull-down.

(b) Configuration de résistance Pull-up

Figure 53 – Configuration des circuits de mesure en utilisant résistances Pull-down et pull-up

$10 M\Omega$, une valeur de $10k$ ne sera pas comparable. De même, nous n'avons qu'une variable à changer, c'est-à-dire un degré de liberté. Ainsi, la conception de circuit sera limitée.

2. Bridge Wheatstone :

L'autre option est d'utiliser un pont de résistances dans une configuration Wheatstone. Cette option permet de mesurer la résistance du capteur FSR en ajustant les trois autres résistances. Dans ce cas, on a trois degrés de liberté pour notre conception. La configuration Wheatstone est illustré ci-dessous, sur la figure 54.

Cette configuration peut s'étudier en utilisant les lois de Kirchhoff ou de contempler le circuit comme deux branches indépendantes, comme le montre la figure 55 et la notion des diviseurs de tension.

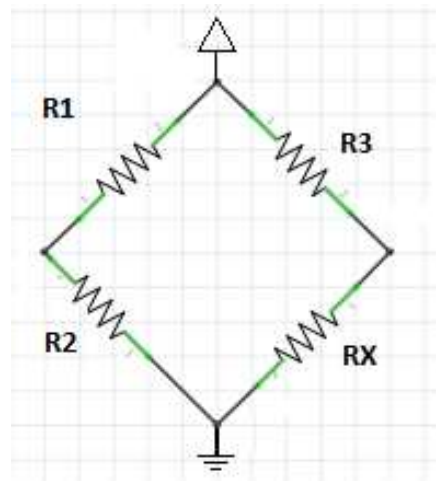


Figure 54 – Configuration de quatre résistances pour le pont Wheatstone

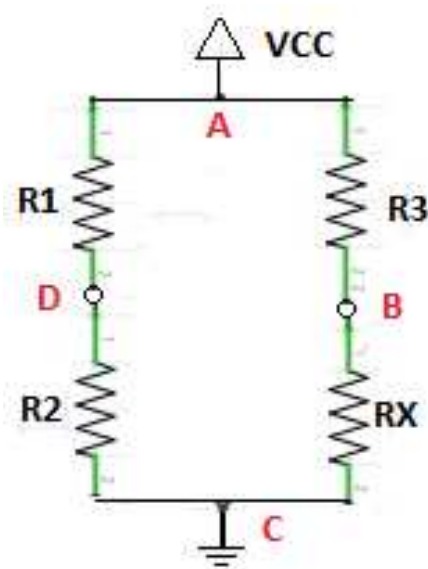


Figure 55 – Diviseurs de tension dans le pont Wheatstone

En premier lieu, nous avons besoin de connaître la tension entre les bornes D et B. La première branche correspond au diviseur de tension des résistances R_1 et R_2 . L'expression se montre dans l'équation 1.

$$V_{B1} = V_{CC} \cdot \frac{R_2}{R_1 + R_2} \quad (1)$$

De manière similaire, l'expression pour la branche qui correspond aux résistances R_3 et R_x (FSR) est affiché ensuite.

$$V_{B2} = V_{CC} \cdot \frac{R_x}{R_x + R_3} \quad (2)$$

Après, si nous joignons les équations (1) et (2) afin d'obtenir une expression pour la tension V_{DB} . Nous obtiendrons l'équation suivante.

$$V_{DB} = V_{CC} \cdot \left(\frac{R_2}{R_1 + R_2} - \frac{R_x}{R_x + R_3} \right) \quad (3)$$

L'équation (3) sera la fonction de transfert de notre pont de Wheatstone et on va l'utiliser pour désigner le circuit. En premier lieu, nous avons les conditions suivantes :

— $V_{CC} = 5V$

— $1K\Omega < R_X < 10M\Omega$

— Il faut que $\frac{R_2}{R_1+R_2} > \frac{R_x}{R_x+R_3}$, afin de garantir toujours des tensions positives.

En tenant en compte les considérations antérieures, on fera le design en étudiant les variations de chaque branche. On commencera avec la branche V_{B2} . L'idée est de trouver la meilleure résistance R_3 , qui donne la plage la plus large possible. Dans la table 2, se montrent les variations par rapport à la tension dans la branche V_{B2} .

Selon les plages obtenues, la valeur de la résistance R_3 qui permet d'obtenir meilleures

R_3	$\frac{R_x}{R_x+R_3}$
1k Ω	0.63 - 0.99
5k Ω	0.33 - 0.95
10k Ω	0.17 - 0.909
15k Ω	0.119 - 0.869
20k Ω	0.092 - 0.833
22k Ω	0.089 - 0.891
33k Ω	0.05 - 0.75
47k Ω	0.041 - 0.68
50k Ω	0.039 - 0.666
70k Ω	0.029 - 0.587
100k Ω	0.019 - 0.508

Table 2 – Planche des variations de la fonction de la branche deux V_{B2} , en changeant la résistance R_3

résultats est 10k Ω , parce qu'elle concède une variation entre 0.17 et 0.95. Cette plage

garantie que la majorité des valeurs soient utilisées sans pertes dans les extrêmes supérieures et inférieures. Le deuxième pas est faire que la fraction $\frac{R_2}{R_1+R_2}$ soit plus grand que le limite supérieur de la branche deux V_{B2} . C'est-à-dire, $\frac{R_2}{R_1+R_2} > 0.909$.

En utilisant les valeurs commerciales des résistances (Table 3), il faut que nous trouvions une paire adéquate.

1	1.2	1.5	1.8	2	2.2	2.7
3.3	3.9	4.7	5.6	6.8	8.2	9.1

Table 3 – Guide des valeurs commerciales des résistances

Après le calcul, les valeurs des résistances R_1 et R_2 sont $1\text{k}\Omega$ et $47\text{k}\Omega$ respectivement.

On doit mesurer les valeurs réelles des résistances qu'on va utiliser pour chacun de capteur afin de connaître le rapport de $\frac{R_2}{R_1+R_2}$. Ensuite, dans la table 4, on montre les valeurs mesures.

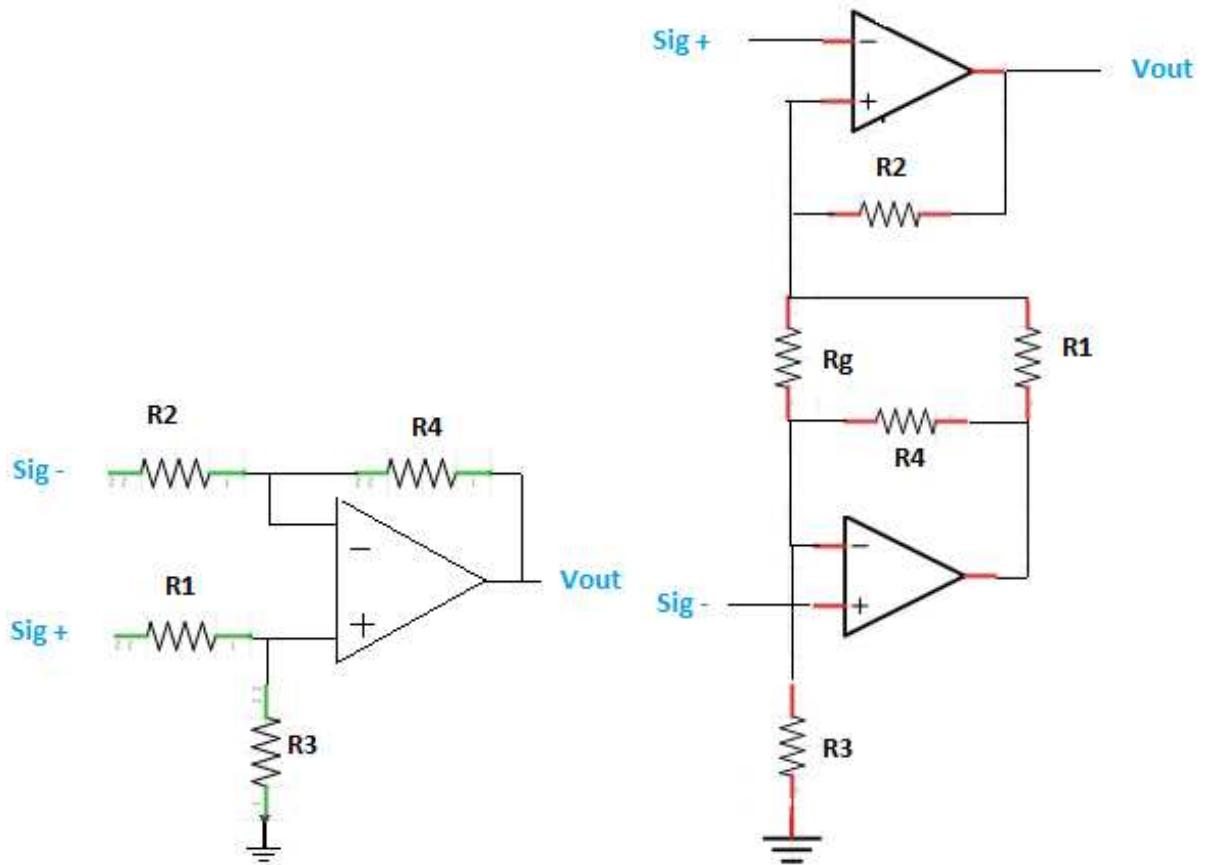
Résistance/Capteur	S1	S2	S3	S4	S5	S6
R_1	987Ω	981Ω	992Ω	990Ω	987Ω	991Ω
R_2	$45.8\text{k}\Omega$	$46.3\text{k}\Omega$	$46.2\text{k}\Omega$	$46.3\text{k}\Omega$	$46.3\text{k}\Omega$	$46.3\text{k}\Omega$
R_3	$9.85\text{k}\Omega$	$9.85\text{k}\Omega$	$9.88\text{k}\Omega$	$9.80\text{k}\Omega$	$9.86\text{k}\Omega$	$9.89\text{k}\Omega$
$\frac{R_2}{R_1+R_2}$	0.9789	0.9792	0.9789	0.9790	0.9791	0.9790

Table 4 – Valeurs réelles des résistances du pont Wheatstone de chaque Capteur.

3. Amplificateur d'instrumentation ou de mesure :

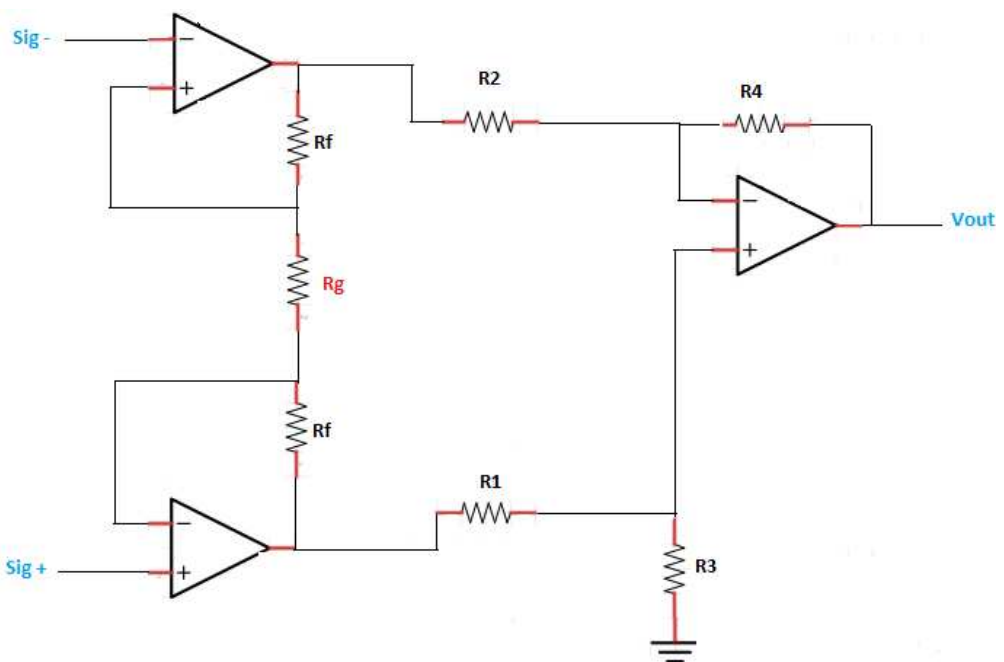
Cette configuration est utilisée afin de traiter de petites signaux et de faire du conditionnement de signaux divers. Dans notre cas, elle nous permettra de traiter et d'amplifier le signal. Il y a différents types d'amplificateur qui sont montrés ensuite.

Dans la figure 56, vous pouvez observer trois configurations différentes. La première se montre dans la figure 56(a), il s'agit d'un schéma qui utilise un amplificateur opérationnel et quatre résistances pour varier le gain du circuit. Cependant, ce circuit donne une impédance d'entrée basse provoquant des changements de courant quand nous connectons certains dispositifs à l'entrée.



(a) Amplifier avec 1 opamp.

(b) Amplifier avec 2 opamps.



(c) Amplifier avec 3 opamps.

Figure 56 – Différentes configurations d'Amplificateurs d'instrumentation

Par contre, les figures 56(b) et 56(c), montrent des configurations en utilisant deux et trois amplificateurs opérationnels, respectivement. Les deux circuits offrent une impédance d'entrée haute qui garantit que si nous connectons un dispositif à l'entrée, le courant ne changera pas trop.

Nous avons choisi, la configuration qui utilise trois amplificateurs opérationnels en tenant compte de deux raisons : Impédance d'entrée haute et circuit équilibré par rapport aux signaux d'entrée [21].

En premier lieu, on fixera le gain en utilisant l'équation suivante :

$$V_{Out} = ((Sig+) - (Sig-)) \times \left(\frac{R4}{R2} \times \left(\frac{2 \bullet Rf}{Rg} + 1 \right) \right) \quad (4)$$

Selon [?], on peut utiliser un paquet des résistances $R_1 - R_4$ d'une même valeur et fixer le gain en utilisant la résistance R_g . Dans ce cas, le gain de l'amplificateur d'instrumentation sera déterminé selon l'équation suivante.

$$A_I = \frac{R4}{R2} \times \left(\frac{2 \bullet Rf}{Rg} + 1 \right) \quad (5)$$

C'est important que nous fixions le gain en tenant compte de la valeur de tension d'entrée et de sortie de chaque dispositif. La figure 57 illustre le schéma général pour le conditionnement des capteurs. On travaillera en prendre en compte la figure 57. Nous connaissons la valeur de la tension d'alimentation de circuit (5 V), et la valeur de $\frac{R2}{R1+R2}$ (Table 4) et la plage pour le rapport $\frac{Rx}{Rx+R3}$, selon les considérations que nous avons montré dans la table 2.

Nous reprenons l'équation 3, et on va remplacer les valeurs comme se montre ensuite.

$$V_{DB} = V_{CC} \bullet \left(\frac{R2}{R1 + R2} \text{moyenne} - \frac{Rx}{Rx + R3} \text{max/min} \right) \quad (6)$$

De cette manière, nous avons deux entrées possibles pour l'amplificateur :

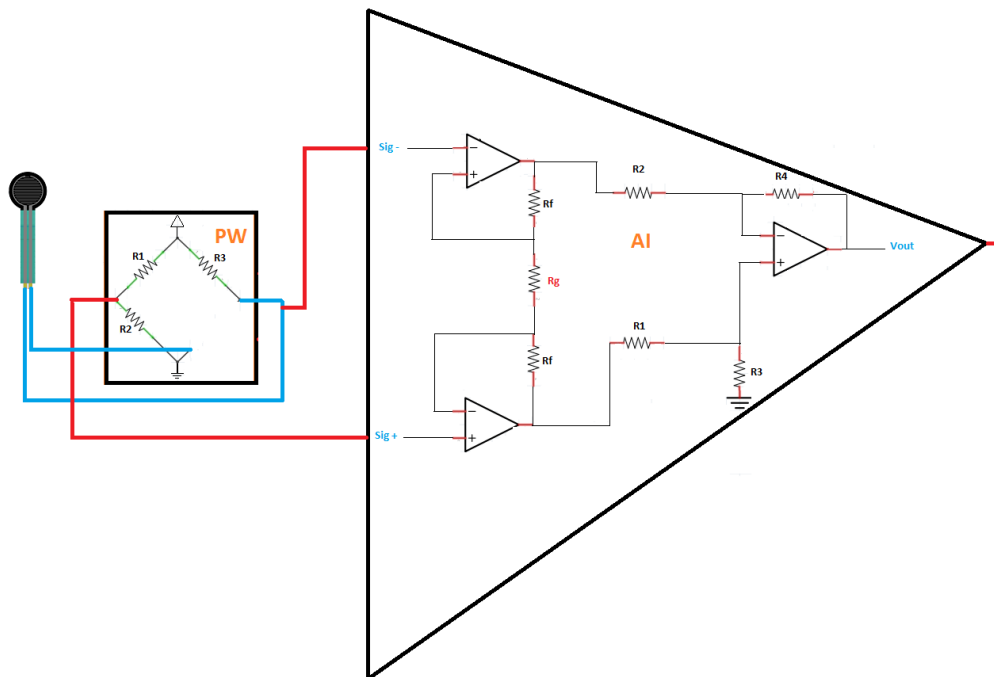
— Le cas 1 :

$$V_{DBmax} = 5V \bullet (0.9790 - 0.17) = 4.045 \quad (7)$$

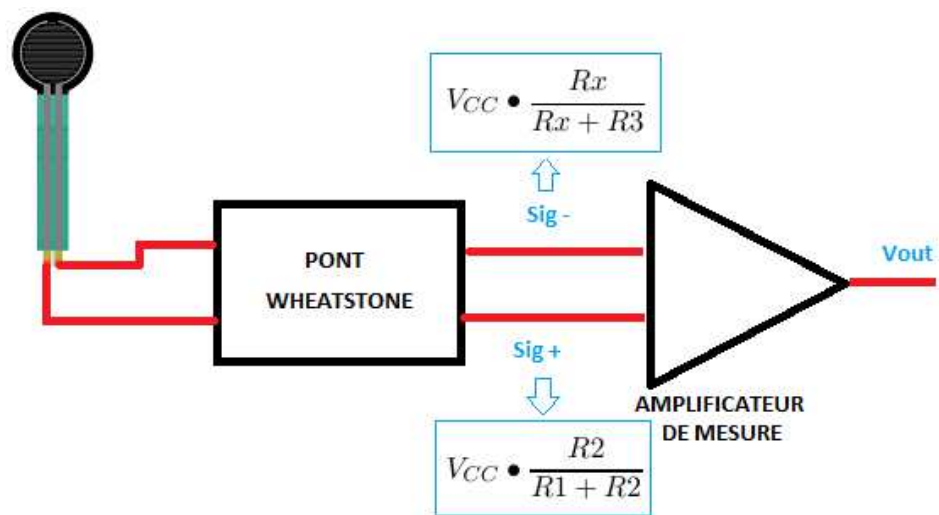
— Le cas 2 :

$$V_{DBmin} = 5V \bullet (0.9790 - 0.909) = 0.35 \quad (8)$$

Nous obtiendrons les valeurs montrées dans les equations (7) et (8). Alors, la sortie peut changer entre 4.045 et 0.35. Pour régler la plage, on va profiter du gain de l'amplificateur de mesure. Pour cette raison, on va considérer un gain de 1.25 à peu près. De cette manière, on travaillera avec une plage de valeurs par rapport à la sortie entre 0.43 et 5. Cette plage facilitera le travail avec un convertisseur analogique-numérique (CAN).



(a) Disposition interne de chaque bloc.



(b) Schéma resumée.

Figure 57 – Schéma général pour le conditionnement des capteurs

4.2.4 Calibrage des capteurs

Les fournisseurs ont fait des essais de calibration en utilisant une barre de 4 mm couverte d'un caoutchouc de 2 mm d'épaisseur.

De notre côté, on a implémenté le système montre dans la figure 58, on a placé des rondelles de la même taille que la surface active des capteurs. L'idée est d'utiliser différents poids sur

la surface de contact et d'activation de nos capteurs, dans notre cas ($37.21mm^2$). Pour faire le calibrage des capteurs, on a utilisé le schéma qui montre la figure 58.

L'objectif de changer de poids et d'établir la courbe de comportement de chaque capteur

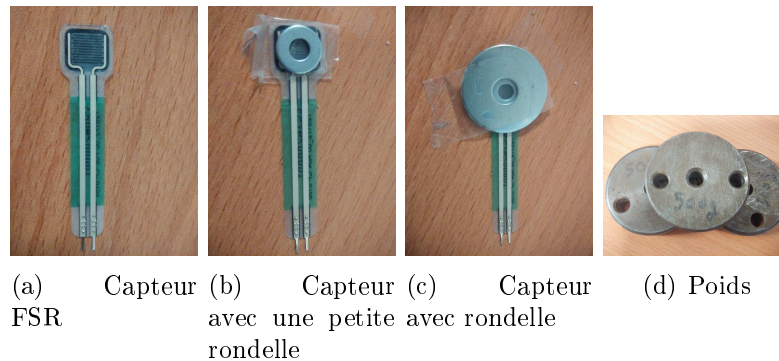


Figure 58 – Calibration des capteurs FSR

comme la figure 52 a montré.

Après des différents essais, on a trouvé que la courbe caractéristique de la force en fonction de la résistance de chaque capteur correspond à l'équation suivante :

$$Force = a * resistcap^b \quad (9)$$

Où,

a et **b** sont constantes obtenus à travers des expériences développées et **resistcap** est la résistance qui donne chaque capteur. Les valeurs sont présentées ici :

$$Force = 175544.430379465 * resistcap^{-1.0764947921} \quad (10)$$

4.2.5 Mise en place des capteurs de pression sur le forceps

La position des capteurs sera définie en prenant en compte le bon fonctionnement et la bonne mesure de la force et la déformation appliquée sur la tête foetale. En premier lieu, les capteurs sont placés comme le montre la figure 59.

Après, l'idée est de faire des essais, afin d'assurer la répétitivité des expériences et la précision des mesures, pour cette raison, on changera la position des capteurs en cherchant le meilleur emplacement.

On a fait des expériences en utilisant le forceps Pajot que montre la figure 59. On a déterminé qu'on peut profiter du comportement de capteurs et de la symétrie des forces exercées sur la tête foetale, comme la figure 60 montre.

Du coup, on a basculé tous les capteurs dans un seul côté, c'est-à-dire, seulement une cuillère aura les capteurs et en plus, dans ce cas, on a utilisé le forceps modèle Levret. La figure 61

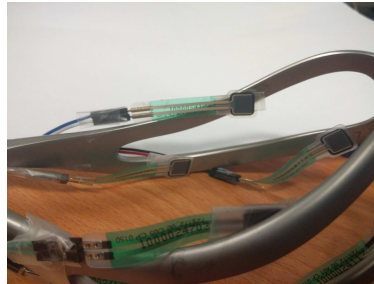


Figure 59 – Disposition initiale des capteurs sur le forceps

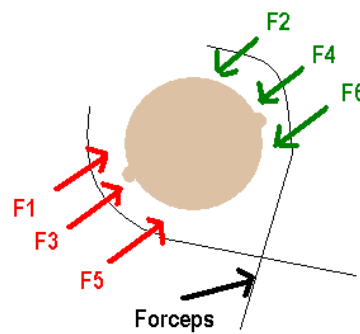


Figure 60 – Forces mesurées pour les capteurs

illustre le changement.

Le placement des capteurs sur un film plastique facilite la mise en place de capteurs sur

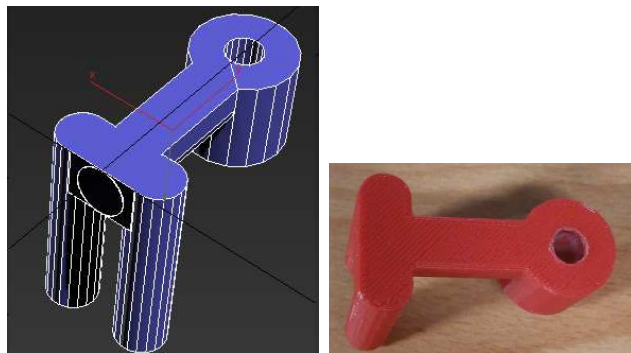


Figure 61 – Disposition de capteurs sur une cuillère de forceps

n'importe quel modèle de forceps.

4.3 Capteur d'écartement

Nous avons développé une pièce afin de fixer les cuillères du forceps et aussi, cette pièce permettra de mesurer l'ouverture du forceps. Dans la figure 62, vous pouvez voir le design.



(a) Design de la pièce en utilisant le logiciel 3DSMax.

(b) Impression 3D.

Figure 62 – Pièce pour fixer les cuillères du forceps

De même, on utilisera un potentiomètre qui nous permettra de connaître l'ouverture du forceps, en utilisant la résistance. Le potentiomètre a une résistance variable entre 0Ω et $10k \Omega$, cependant la variation sera entre 0Ω et $1.5 k \Omega$.

4.3.1 Mise en place de capteur

Cette pièce sera fixée au milieu, dans la partie ronde, comme le montre la figure 63 ci-dessous.

D'autre côté, on va mesurer l'écartement des cuillères du forceps par rapport à la résistance donnée par le potentiomètre.

La figure 64, montre le circuit utilisé afin de récupérer la valeur de la résistance.

4.4 Récupération et traitement des données :

4.4.1 Caractéristiques de la Carte Arduino

Après l'étape de conditionnement, il faut que nous récupérions les données afin d'analyser l'information que donne chaque capteur. À cette effet, nous utiliserons une carte *Arduino UNO*. Cette carte a les caractéristiques suivantes (Figure 65) :

- Alimentation : port USB ou connexion externe.
- Tension de fonctionnement : 5V

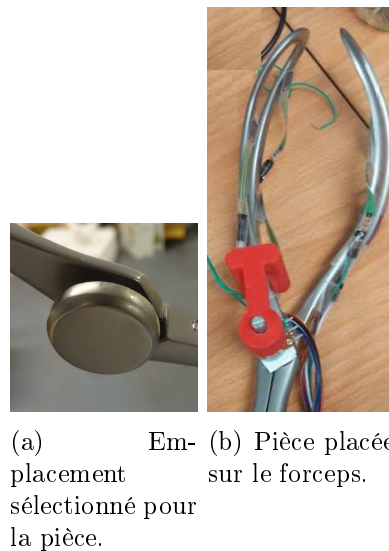


Figure 63 – Disposition de la pièce sur le forceps

- 6 entrées analogiques 10 bits
- 16 entrées/sorties numériques

4.4.2 Connexion des Capteurs de pression sur la carte Arduino

Pour notre application, on branchera chaque sortie de nos amplificateurs de mesure aux entrées analogiques que fournit Arduino Uno [22], comme la figure 66 l'illustre.

Dans la figure 66, PW est l'abréviation pour Pont de Wheatstone et AI correspond à Amplificateur d'Instrumentation.

4.4.3 Connexion de Capteur d'écartement sur la carte Arduino

Dans le cas de potentiomètre, on a utilisée aussi la configuration de Pont Wheatstone, que nous avons exprimée dans la section 4.3.1. Cependant, on a ajouté une petite résistance, afin d'assurer que la mesure de la tension V_{db} soit toujours supérieure à zero, comment la figure 64 montre.

On utilisera l'entrée analogique numéro 5, afin de récupérer la valeur de la résistance, comme l'illustre la figure 67.

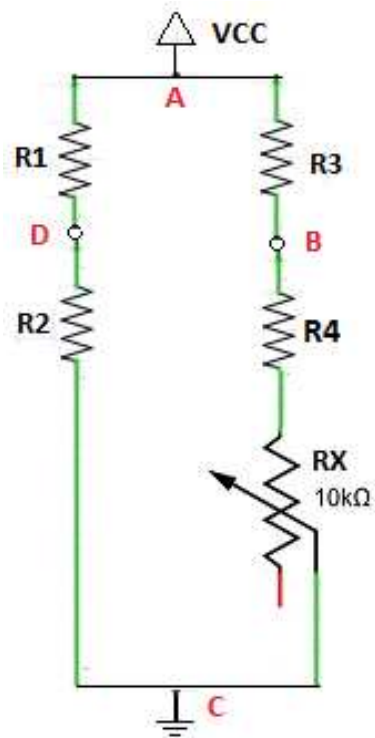


Figure 64 – Schéma utilisée pour récupérer la valeur de la résistance

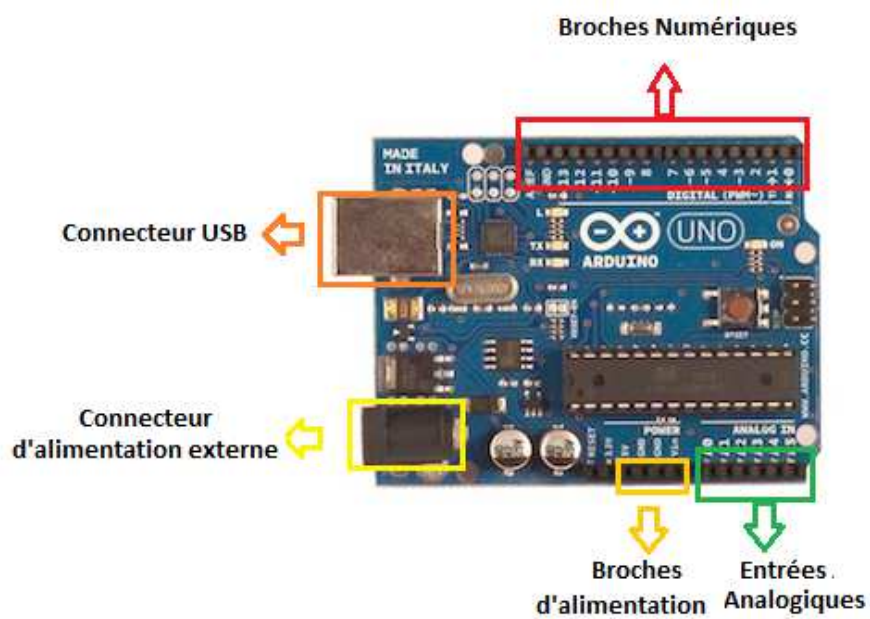


Figure 65 – Disposition des broches dans la Carte Arduino Uno. Adaptée de [22]

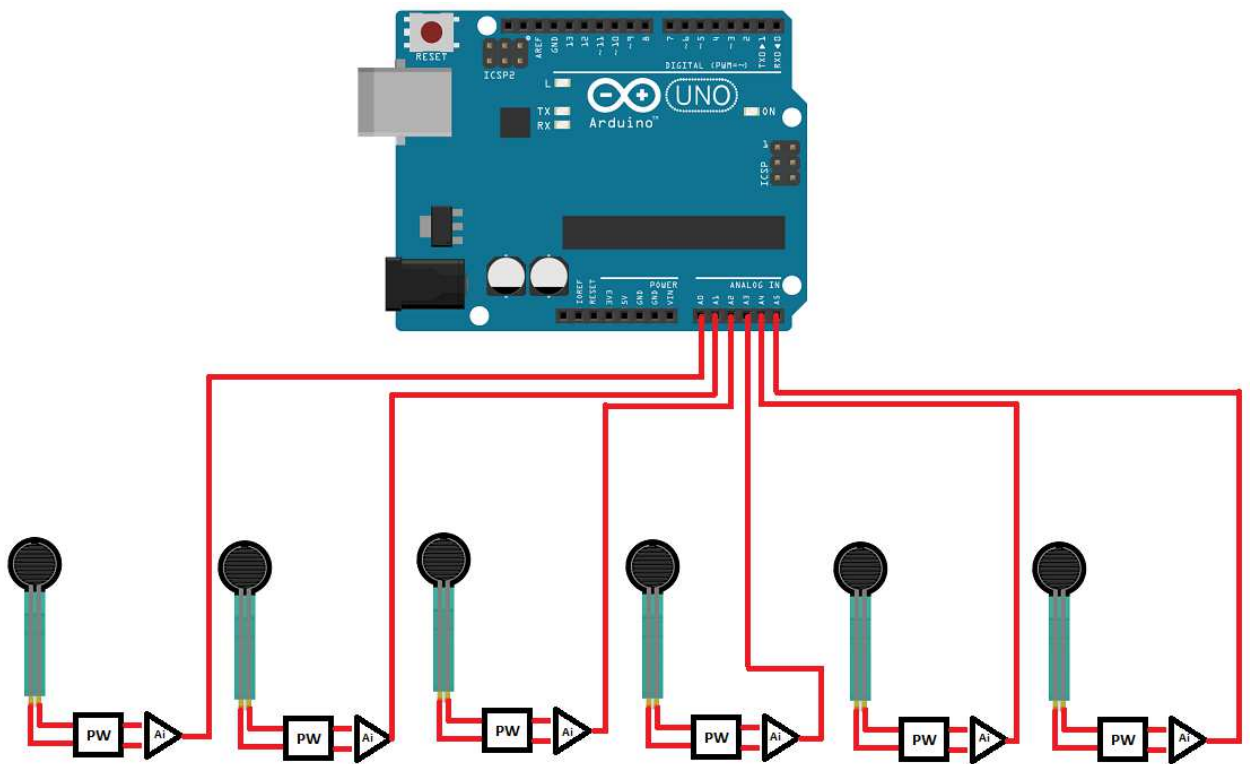


Figure 66 – Schéma général de connexion pour le conditionnement des capteurs et récupération des données

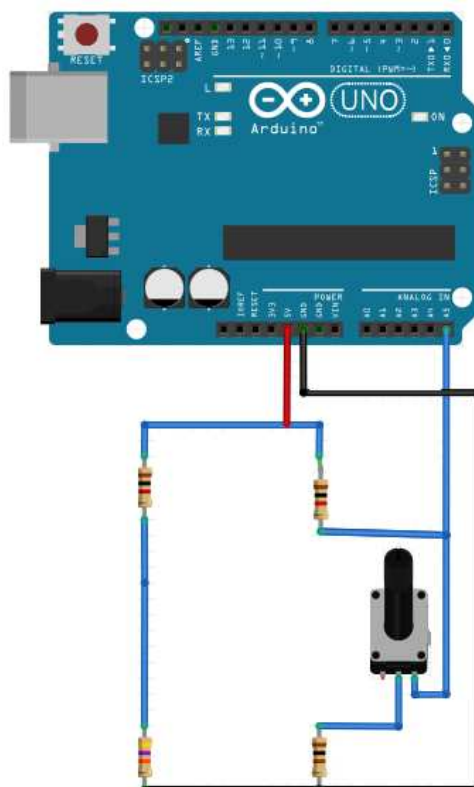


Figure 67 – Schéma pour récupérer la valeur de la résistance de potentiomètre

4.4.4 Codage sur Arduino

De même, c'est nécessaire de commander la carte *Arduino UNO* en utilisant l'ordinateur. Il y a un logiciel qui s'appelle *Arduino* (disponible depuis sa page officielle : <https://www.arduino.cc/en/Main/Software>) qui permet de faire la programmation de la carte en utilisant des commandes faciles. Aussi, la structure des codes dans le logiciel Arduino suivi le schéma suivant :

```
//Définition des variables
void setup()
{
  // ...
}
void loop()
{
  // ... }
```

Les codes développés sur Arduino, se montrent dans la section Annexes.

4.5 Interface Graphique

Pour faire des analyses des données de manière plus facile, on a développé une interface graphique qui permet d'observer les changements de capteurs de pression.

Dans ce cas, on a utilisé le logiciel Processing (disponible depuis sa page officielle : <https://processing.org/> que facilite le développement de plusieurs applications en 2D et 3D. Ce logiciel est disponible sur les systèmes d'exploitation Windows, Mac et Linux.

Dans la figure 68, on peut regarder l'interface créée en utilisant Processing.

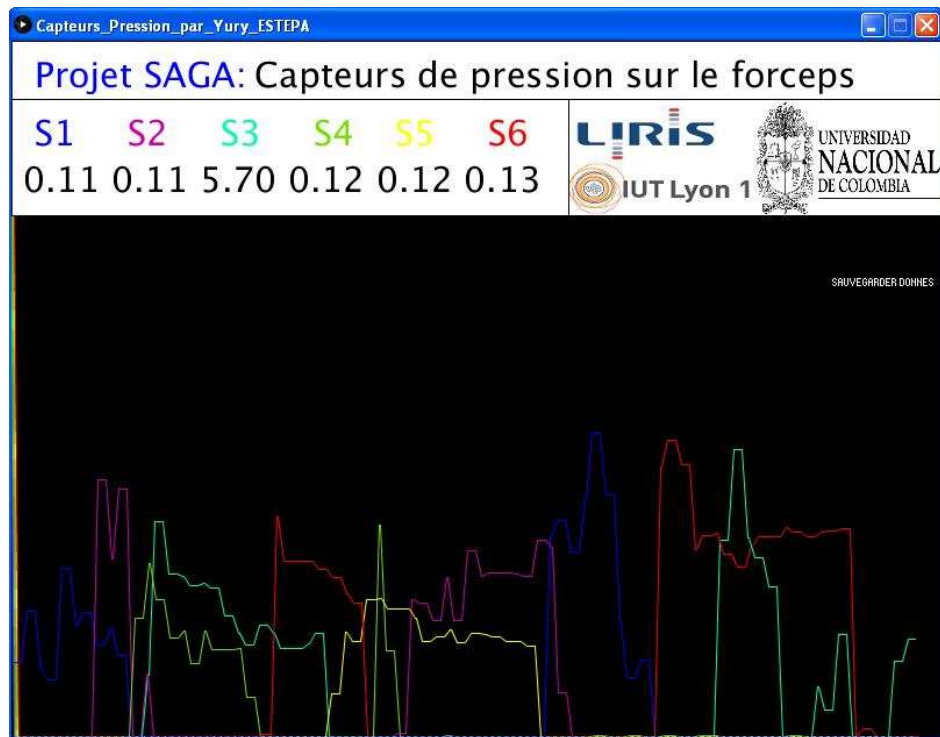


Figure 68 – Interface pour la visualisation des capteurs de pression

Nous pouvons vérifier le fonctionnement de chaque capteur et on a la possibilité de sauvegarder les données dans un fichier de texte. Quand on appuie sur cette option on crée un fichier qui s'appelle *sensores.txt*. Avec le fichier de texte, nous pouvons gérer les données selon nos besoins. Par exemple, on a décidé de faire l'exportation des données au logiciel Microsoft Office Excel.

5 Conclusions et travaux futurs

- En utilisant le logiciel 3DS Max 2015 d'Autodesk, on peut travailler de manière facile quand on utilise des maillages polygonaux. Cependant, l'outil de conversion de maillages polygonaux afin d'obtenir NURBS génère beaucoup des erreurs comme la perte des caractéristiques physiques du maillage.
- Le NURBS que génère 3DS Max n'est pas exploitable. Il génère un maillage en mosaïque à partir des polygones originaux. Il faut utiliser d'autre outil si nous avons besoin de travailler avec Nurbs.
- La conversion du maillage en utilisant Rhino offre beaucoup des options afin d'obtenir une surface. Cependant, il y a le risque d'obtenir des surfaces qui ont les normales inversées. Ce problème provoque des soucis dans les traitements suivants, par exemple, dans la création de tétraèdres. C'est toujours nécessaire que nous changions les orientations des normales.
- C'est possible d'ajouter des capteurs de pression additionnels, afin de mesurer de manière complète les forces exercées par le forceps dans la tête foetale. Cependant, il faut choisir une carte Arduino avec plusieurs entrées analogiques ou d'autre carte électronique plus puissante.
- Les forces qui sont mesurées en utilisant les capteurs de pression ont été considérées symétriques dans chaque cuillère du forceps, afin de faciliter les expériences et de pouvoir mesurer dans différents points.
- Quand la force de chaque capteur et l'écartement ont été mesurées, on peut faire la supposition que la somme de chacun des forces sera la force totale que la personne a exercée en utilisant le forceps sur la tête foetale.
- C'est intéressant aussi d'obtenir les forces qui donnent les capteurs et faire une comparaison avec les forces que le simulateur Birthsim obtienne. Afin de réaliser cette procédure, il faut que on tienne en compte les déformations produites sur la tête foetale. Pour cette raison, c'est nécessaire si nous pouvons parler d'un travail (force * distance).

6 Annexes

6.1 Annexe A. Code Arduino

```
/*Definicion de variables*/
int sensor1,sensor2,sensor3,sensor4,sensor5,sensor6;
//usar estas dos variables si se conectan los sensores 7 y 8
int sensor7,sensor8;
float Tension_sensor1,Tension_sensor2,Tension_sensor3,Tension_sensor4;
float Tension_sensor5,Tension_sensor6;
//usar estas dos variables si se conectan los sensores 7 y 8
float Tension_sensor7,Tension_sensor8;
int alimentacion=5000,j; //5V Valor en mV
//Puente Wheatstone Sensor 1
int R11=987;/* 1k*/
int R21=45800;/* 47k*/
int R31=9850;/* 10k*/
//Puente Wheatstone Sensor 2
int R12=981;/* 1k*/
int R22=46300;/* 47k*/
int R32=9850;/* 10k*/
//Puente Wheatstone Sensor 3
int R13=992;/* 1k*/
int R23=46200;/* 47k*/
int R33=9880;/* 10k*/
//Puente Wheatstone Sensor 4
int R14=990;/* 1k*/
int R24=46300;/* 47k*/
int R34=9800;/* 10k*/
//Puente Wheatstone Sensor 5
int R15=987;/* 1k*/
int R25=46300;/* 47k*/
int R35=9860;/* 10k*/
//Puente Wheatstone Sensor 6
int R16=991;/* 1k*/
int R26=46300;/* 47k*/
int R36=9890;/* 10k*/
//Puente Wheatstone Sensor 7
int R17=990;/* 1k*/
int R27=46300;/* 47k*/
```

```
int R37=9850; /* 10k*/
//Puente Wheatstone Sensor 8
int R18=986; /* 1k*/
int R28=46200; /* 47k*/
int R38=9870; /* 10k*/
//Puente Wheatstone Potenciometro
int R17=995; /* 1k*/
int R27=46400; /* 47k*/
int R37=997; /* 1k*/
float A,B,C,D,E,F,V;
//Declaración Resistencia sensores
float Res_sensor1,Res_sensor2,Res_sensor3,Res_sensor4,Res_sensor5,Res_sensor6;
float Res_pot6;
//usar estas dos variables si se conectan los sensores 7 y 8
float Res_sensor7,Res_sensor8;
float Fuerza_Tot,separacion[100], deformacion[100],Travail[100];
//Declaración Fuerza sensores
double fuerza1,fuerza2,fuerza3,fuerza4,fuerza5,fuerza6,Res_pot6;
//usar estas dos variables si se conectan los sensores 7 y 8
double fuerza7,fuerza8;
//Constantes encontradas por interpolacion de la curva caracteristica de los
//sensores
float cons=175544.430379465;
float cons1=-1.0764947921;

void setup(void)
{
  Serial.begin(9600);
}
void loop (void)
{
  sensor1 = analogRead(A0);
  delay(2);
  sensor2 = analogRead(A1);
  delay(2);
  sensor3 = analogRead(A2);
  delay(2);
  sensor4 = analogRead(A3);
  delay(2);
  sensor5 = analogRead(A4);
```

```
delay(2);
sensor6 = analogRead(A5);
delay(2);
//usar estas dos variables si se conectan los sensores 7 y 8
//sensor7 = analogRead(A6);
//delay(2);
//sensor8 = analogRead(A7);
//delay(2);

Tension_sensor1=map(sensor1,0,1023,0,alimentacion);
Tension_sensor2=map(sensor2,0,1023,0,alimentacion);
Tension_sensor3=map(sensor3,0,1023,0,alimentacion);
Tension_sensor4=map(sensor4,0,1023,0,alimentacion);
Tension_sensor5=map(sensor5,0,1023,0,alimentacion);
Tension_sensor6=map(sensor6,0,1023,0,alimentacion);
//usar estas dos variables si se conectan los sensores 7 y 8
//Tension_sensor7=map(sensor6,0,1023,0,alimentacion);
//Tension_sensor8=map(sensor6,0,1023,0,alimentacion);
A= (Tension_sensor1/alimentacion)-(R21/(R11+R21));
B= (Tension_sensor2/alimentacion)- (R22/(R12+R22));
C= (Tension_sensor3/alimentacion)- (R23/(R13+R23));
D= (Tension_sensor4/alimentacion)- (R24/(R14+R24));
E= (Tension_sensor5/alimentacion)- (R25/(R15+R25));
F= (Tension_sensor6/alimentacion)- (R26/(R16+R26));
//usar estas dos variables si se conectan los sensores 7 y 8
//G= (Tension_sensor7/alimentacion)- (R27/(R17+R28));
//H= (Tension_sensor8/alimentacion)- (R28/(R18+R28));
V= (Tension_sensor6/alimentacion);
Res_sensor1=- (R31*A)/(1+A);
Res_sensor2=- (R32*B)/(1+B);
Res_sensor3=- (R33*C)/(1+C);
Res_sensor4=- (R34*D)/(1+D);
Res_sensor5=- (R35*E)/(1+E);
Res_sensor6=- (R36*F)/(1+F);
//usar estas dos variables si se conectan los sensores 7 y 8
//Res_sensor7=- (R37*G)/(1+G);
//Res_sensor8=- (R38*H)/(1+H);
Res_pot6=(R37*V)/(1-V);
fuerza1=cons*pow(Res_sensor1,cons1);
fuerza2=cons*pow(Res_sensor2,cons1);
```

```
fuerza3=cons*pow(Res_sensor3,cons1);
fuerza4=cons*pow(Res_sensor4,cons1);
fuerza5=cons*pow(Res_sensor5,cons1);
fuerza6=cons*pow(Res_sensor6,cons1);
//usar estas dos variables si se conectan los sensores 7 y 8
fuerza7=cons*pow(Res_sensor7,cons1);
fuerza8=cons*pow(Res_sensor8,cons1);
// Imprimir valores de las fuerzas medidas
  Serial.print(fuerza1);
  Serial.print(" ");
  Serial.print(fuerza2);
  Serial.print(" ");
  Serial.print(fuerza3);
  Serial.print(" ");
  Serial.print(fuerza4);
  Serial.print(" ");
  Serial.print(fuerza5);
  Serial.print(" ");
  Serial.print(fuerza6);
  Serial.println(" ");
  //usar estas dos variables si se conectan los sensores 7 y 8
  //Serial.print(fuerza7);
  //Serial.println(" ");
  //Serial.print(fuerza8);
  //Serial.println(" ");
  delay(300);
}
```


6.2 Annexe B. Code Processing

```
//declaración librerias
import processing.serial.*;
import controlP5.*;
//Declaracion de variables
PrintWriter datos;
Serial PuertoArduino;    // Puerto serial
String inString; // Input desde Puerto serial
int lf = 10;           // ASCII salto de linea
float x1 = 0;
float x2;
float y1;
float y2 = height / 2;
int xPos = 1;
PImage foto, foto1;
String[] lista;
float[] listaY = new float[10];
// Agregar color si se grafican más de 6 valores
int[] colores = { #0000FF, #CC00AA, #00FFAA, #74DF00, #FFFF00, #FF0000, #F7FE2E};
ControlP5 cp5;
void setup()
{
    //Tamaño de la Ventana
    size(800, 600);
    cp5 = new ControlP5(this);

    cp5.addButton("Sauvegarder donnees")
        .setValue(0)
        .setPosition(700,200)
        .setSize(100,15)
        .setColorBackground(color(0, 0, 0))
        ;
    // Imprime todos los puertos disponibles
    printArray(Serial.list());
    //Para este caso, Serial.list()[1], SE DEBE CAMBIAR ESTE VALOR
    // POR EL PUERTO CORRESPONDIENTE
    PuertoArduino = new Serial(this, Serial.list()[1], 9600);
    PuertoArduino.bufferUntil(lf);
    //crea un archivo txt
```

```
datos = createWriter("sensores.txt");
background(0);// color de fondo
textSize(30);// Tamaño del texto
//Cargar logos de las universidades
foto = loadImage("liris2.png");
foto1 = loadImage("unal.jpg");
}
void draw()
{
  if (inString != null)
  {
    // Remueve los espacios en blanco:
    inString = trim(inString);
    //Espacio para mostrar los datos
    fill(255);
    stroke(0);
    rect(0, 0, 800, 50);
    fill(255);
    stroke(0);
    rect(0, 50, 479, 100);
    //Lectura de cada sensor
    fill(0, 0, 255);
    text("S1", 20, 90);
    fill(204, 0, 170);
    text("S2", 100, 90);
    fill(0, 255, 170);
    text("S3", 180, 90);
    fill(116, 223, 0);
    text("S4", 260, 90);
    fill(255, 255, 0);
    text("S5", 330, 90);
    fill(255, 0, 0);
    text("S6", 410, 90);
    fill(0, 0, 0);
    text(inString, 10, 130);
    fill(0, 0, 255);
    text("Projet SAGA:", 20, 40);
    fill(0, 0, 0);
    text(" Capteurs de pression sur le forceps", 200, 40);
    image(foto, 480, 51, 160, 99);
```

```
image(foto1, 640, 51, 160, 99);
// Separa los valores recibidos usando espacio
lista = split(inString, ' ');
// Mostrar lista
printArray(lista);
// Recorre la lista de valores
for (int i=0; i < lista.length; i++)
{
    // Convierte el valor en su representacion numerica ( Float )
    float inByte = float(lista[i]);
    datos.print(inByte+","); // copia el dato en sensores.txt
    // Recupera el valor guardado de la iteracion anterior para usarlo
    //como coordenada
    y1 = listaY[i];
    // mapeo segun tamaño de la pantalla
    inByte = map(inByte, 0, 40, 0, height);
    // dibujar la linea
    stroke(colores[i]);
    y2 = height - inByte;
    line(xPos - 5, y1, xPos, y2);
    // Guarda la coordenada y actual
    listaY[i] = y2;
    // cuando se acabe la linea, se vuelve al comienzo de la pantalla
    if (xPos >= width)
    {
        xPos = 0;
        background(0);
    }
    else
    {
        // incremento en la posición horizontal
        xPos+=1;
    }
}
datos.println(" "); //
// Sincronizar el retardo con el programa que envia
//los valores al puerto serial
delay(300);
}
}
```

```
void serialEvent(Serial p)
{
    inString = p.readString();
}
public void controlEvent(ControlEvent theEvent)
{//presionar botón para guardar y salir
    println(theEvent.getController().getName());
    datos.flush();
    datos.close();
    exit();
}
void keyPressed()
{//Presionar cualquier tecla para guardar y salir
    datos.flush();
    datos.close();
    exit();
}
```

6.3 Annexes C. Carte de Prototypage

Vous pouvez trouver dans la figure 69, la disposition des capteurs dans la carte de prototypage. En sachant que S1 correspond au capteur 1, etc. Ici, il y a huit capteurs pour donner la possibilité d'obtenir plusieurs mesures. Il faut, alors, qu'on utilise une carte Arduino avec plusieurs entrées analogiques ou d'autre carte électronique afin de récupérer les valeurs de chaque capteur. Si vous voulez faire de nouveau les expériences, pour vous-même, il faut que

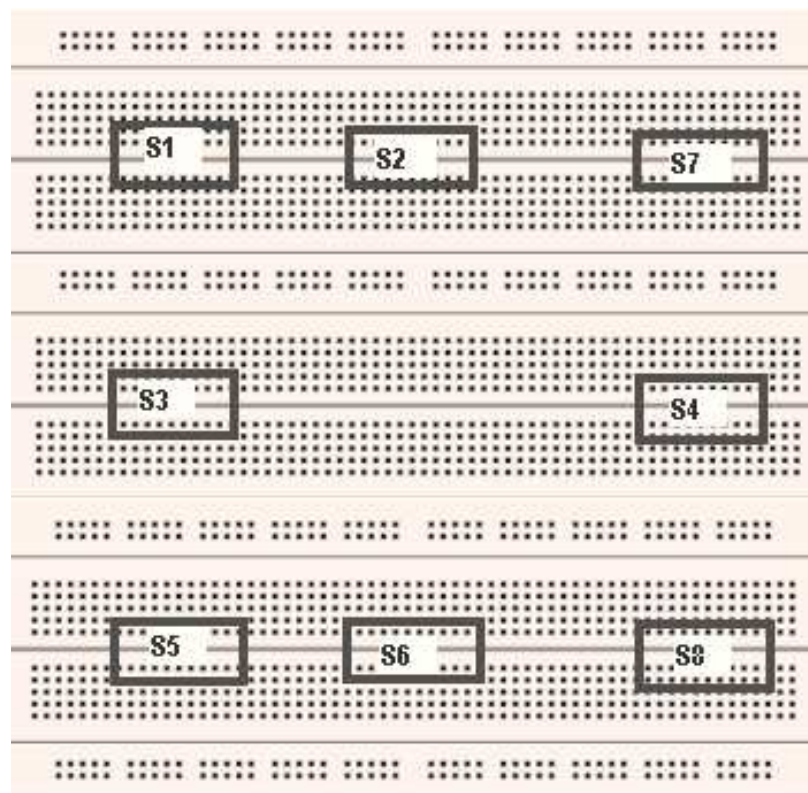


Figure 69 – Disposition de composants électroniques pour chaque capteur sur la carte de prototypage

vous teniez en compte les conseils suivants :

- Suivre le schéma électronique de la figure 70. De même, dans la figure 71, vous pouvez trouver les connexions faites pour chaque capteur.
- Les connexions doivent être vérifiées, afin de voir que le circuit fonctionne de manière correcte.
- Il y a deux sources d'alimentation :
 - 5V (Sortie Arduino) : Cette tension est utilisée pour tous les circuits Pont de Wheatstone.
 - 6.3 V (Source d'alimentation externe) : Cette tension est utilisée pour les circuits de conditionnement (chip MCP6004)

Pour cette raison, c'est important de ne pas mélanger l'alimentation, parce que la conception du circuit changera et les valeurs de tension de sortie pour chaque capteur ne seront pas appropriées.

- C'est important que vous n'oubliez pas brancher les masses. Les deux masses doivent être branchées. Afin d'éviter des pertes dans le circuit que peuvent perturber le fonctionnement des capteurs.
- Vous devez respecter les valeurs des résistances afin de garantir le bon fonctionnement du circuit.

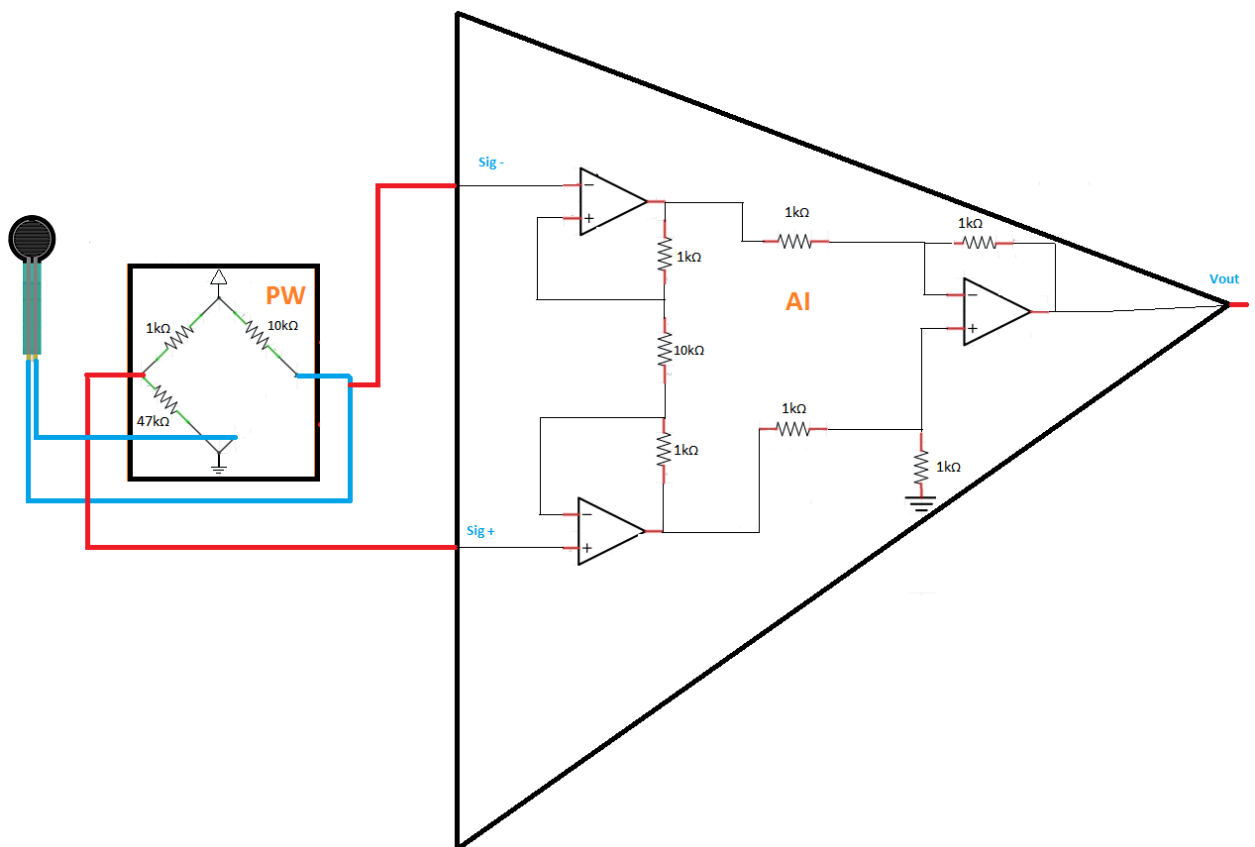


Figure 70 – Schéma électronique du circuit implementé

6.4 Annexes D. Couplage Dspace Birthsim

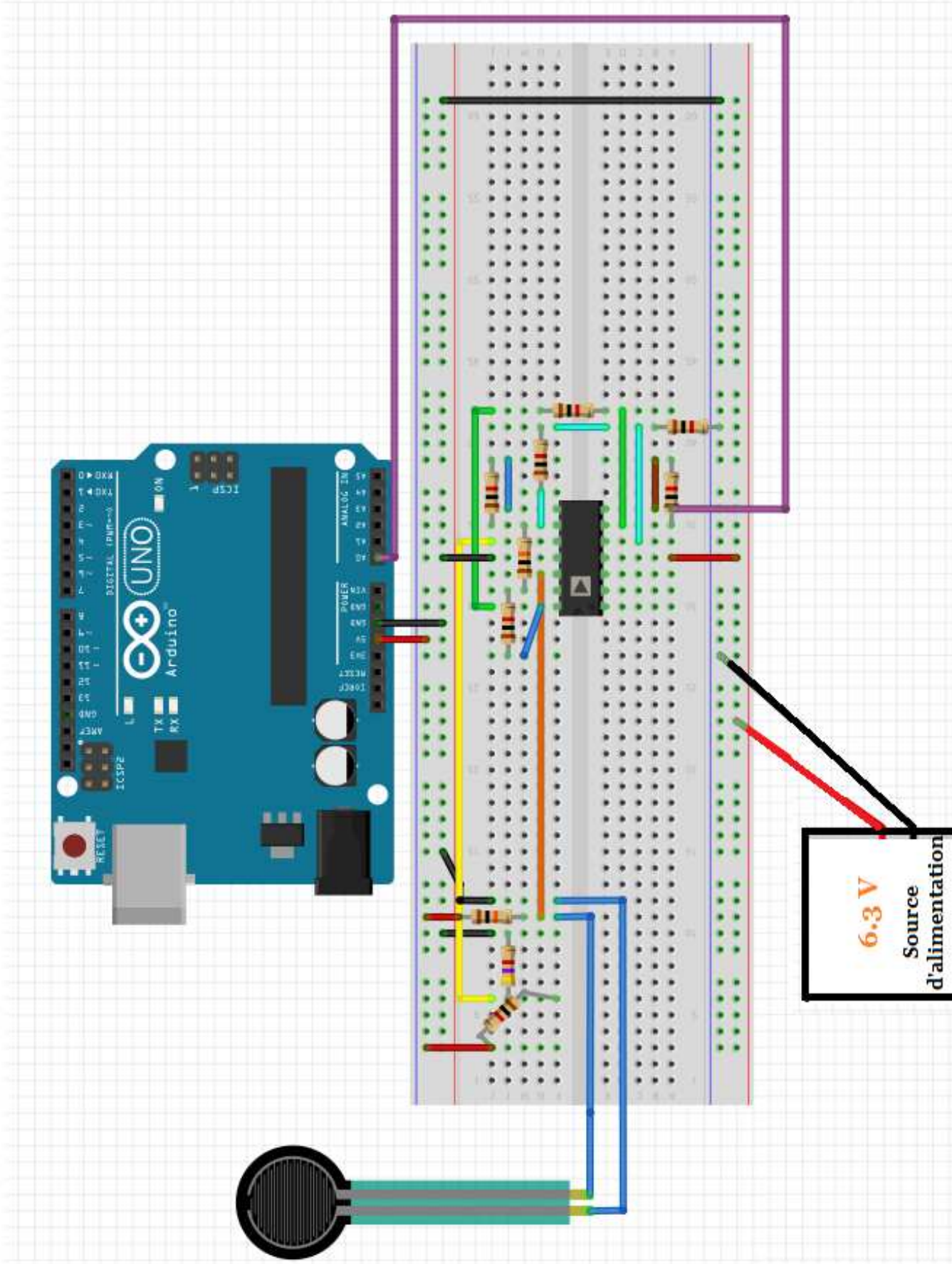
Si vous êtes intéressée par faire la connexion de ce projet au Simulateur Birthsim. Vous pouvez profiter et vous pouvez faire des connexions directement en utilisant les câbles VNC qui se trouvent déjà branchés à Dspace. Il faut que vous suiviez les pas suivants :

- Vous devez brancher les huit câbles aux sorties de chaque capteur en suivant le code de couleur qui se montre ensuite : Ou vous pouvez suivre le schéma qui se montre dans la figure 72.

S1	Jaune
S2	Noir
S3	Café
S4	Violette
S5	Orange
S6	Bleu
S7	Blanc
S8	Rouge

Table 5

- Vous devez brancher la masse des câbles VNC à la masse de votre carte de prototypage Arduino.



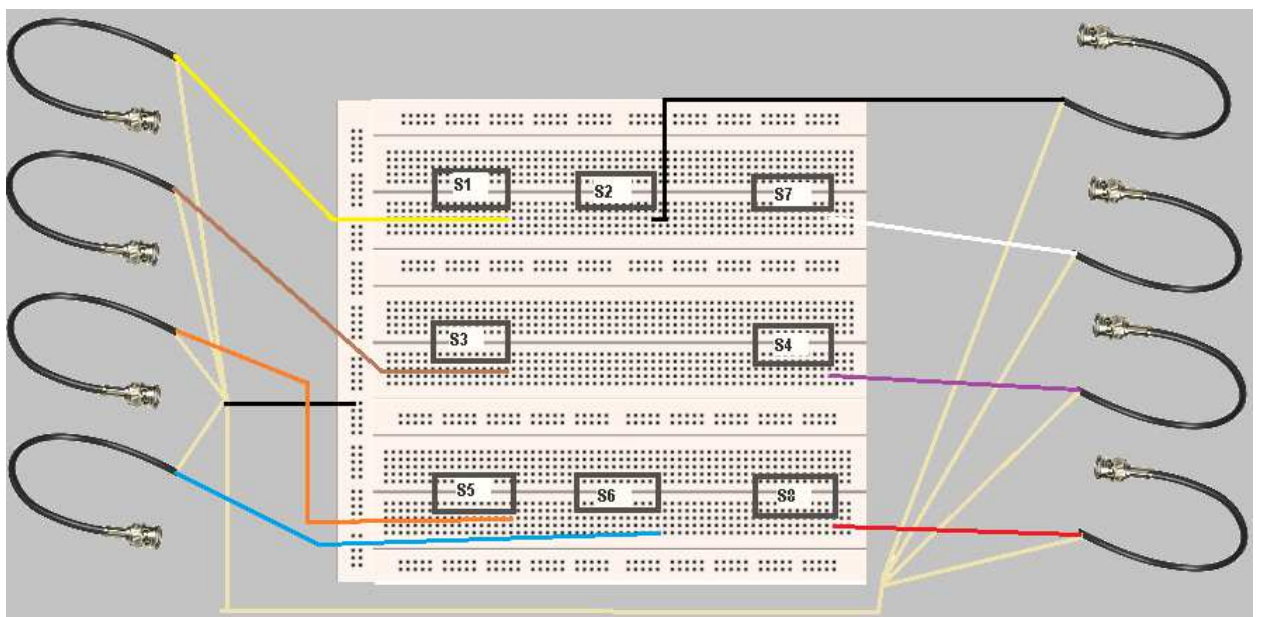


Figure 72 – Connexion de câbles depuis huit câbles de type VNC, pour chaque capteur sur la carte de prototypage

Références

- [1] Jacques Lansac, Philippe Descamps, and Jean-François Oury. *Pratique de l'accouchement*. Elsevier Masson, January 2012. Google-Books-ID : ph7uUtiBdk8C.
- [2] Norma Beyer. Creative educational methodologies : Using a childbirth simulator with baccalaureate nursing students. *Newborn and Infant Nursing Reviews*, 9(2) :88–90, June 1996.
- [3] L P Halamek, D M Kaegi, D M Gaba, Y A Sowb, B C Smith, B E Smith, and S K Howard. Time for a new paradigm in pediatric medical education : teaching neonatal resuscitation in a simulated delivery room environment. *Pediatrics*, 106(4) :E45–E45, October 2000.
- [4] D Korosec, A Holobar, M Divjak, and D Zazula. Building interactive virtual environments for simulated training in medicine using VRML and java/JavaScript. *Computer Methods and Programs in Biomedicine*, 80 :S61–S70, December 2005.
- [5] Kathleen R. Rosen. The history of medical simulation. *Journal of Critical Care*, 23(2) :157–166, June 2008.
- [6] S Abrahamson, J S Denson, and R M Wolf. Effectiveness of a simulator in training anesthesiology residents. 1969. *Quality & Safety In Health Care*, 13(5) :395–397, October 2004.
- [7] R. Arne, F. Stale, K. Ragna, and L. Petter. PatSim - simulator for practising anaesthesia and intensive care. *PatSim - PatSim - Ein Übungssimulator für Anästhesie und Intensivpflege*, (3) :147, 1996.
- [8] Caractéristiques Simulateur SIM NewBaby. Laerdal Medical. <http://www.laerdal.com/fr/doc/88/SimNewB>, 2016.
- [9] ANR MN 2012 - Projet SAGA- WIKI LIRIS. Un dispositif original de simulation pour la formation des sages-femmes et des obstétriciens. <https://liris.cnrs.fr/saga/wiki/lib/exe/fetch.php?media=flyersaga.pdf>.
- [10] Autodesk. 3DS Max 2015 Help. <http://help.autodesk.com/view/3DSMAX/2015>, 2016.
- [11] Rhinoceros. Rhinoceros Help. <http://docs.mcneel.com/rhino/5/help/en-us/index.htm>, 2016.
- [12] Sourceforge. Meshlab. <http://meshlab.sourceforge.net/>, 2016.
- [13] Sourceforge. Netgen. <https://sourceforge.net/projects/netgen-mesher/?source=navbar>, 2016.
- [14] Masson Melchor J. Merger R, Levy J. Précis d'obstetrique. *Paris : Masson*, (6) :624, 1995.
- [15] Ruth Richardson. From the medical museum. *The Lancet*, 358, 2001.

-
- [16] Dumont M. History and sidelights on the forceps. *Journal de Gynecologie, obstetrique et biologie de la reproduction*, 7 :743–747, 1984.
- [17] Shute W.B. An obstetrical forceps using a new principle of parallelism. *Proceedings of the royal society of medicine*, 51(10) :837–839, 1958.
- [18] Louis. Devraigne. L'obstétrique à travers les ages. *Edition Doin*, 1939.
- [19] Rosenberg P. Manoeuvres obstétricales. *Paris : Masson*, page 39, 1993.
- [20] IEE Solution. Customized Input Sensing- CIS Solutions. http://www.produktinfo.conrad.com/datenblaetter/00000-24999/001008981-da-01-en-DRUCKSENSOR_HD001AS.pdf, 2016.
- [21] Texas Instruments. Application Report. Signal Conditioning Wheatstone Resistive Bridge Sensors. <http://www.ti.com/lit/an/sloa034/sloa034.pdf>. SLOA034 - September 1999.
- [22] Arduino. Caractéristiques Carte Arduino UNO. <https://www.arduino.cc/en/Main/ArduinoBoardUno>, 2016.