



HAL
open science

Stéréotomie et vision artificielle pour la construction robotisée de structures maçonnées complexes

Vianney Loing

► **To cite this version:**

Vianney Loing. Stéréotomie et vision artificielle pour la construction robotisée de structures maçonnées complexes. Génie civil. Université Paris-Est, 2019. Français. NNT : 2019PESC1015 . tel-02385841

HAL Id: tel-02385841

<https://pastel.hal.science/tel-02385841>

Submitted on 29 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale n° 531 : Sciences, ingénierie et environnement

THÈSE

Thèse présentée en vue de l'obtention du grade de

Docteur de l'Université Paris-Est

Spécialité doctorale "Mécanique, Génie mécanique, Génie civil"

présentée par

Vianney LOING

Stéréotomie et vision artificielle pour la construction robotisée de structures maçonnées complexes

Thèse soutenue le 22 janvier 2019

Directeur de thèse : **Jean-François CARON**
Co-directeur de thèse : **Renaud MARLET**
Co-encadrant de thèse : **Mathieu AUBRY**
Co-encadrant de thèse : **Olivier BAVEREL**

Jury

M. Corentin FIVET,	Professeur, SXL, EPFL	Rapporteur
M. Vincent LEPETIT,	Professeur, LaBRI, Université de Bordeaux	Rapporteur
Mme. Diane LARLUS,	Senior Research Scientist, NAVER LABS Europe	Examinatrice
Mme. Antonella MASTRORILLI,	Professeur, ENSAP Lille	Présidente

Résumé

Ce travail de thèse s’inscrit dans le contexte du développement de la robotique dans la construction. On s’intéresse ici à la construction robotisée de structures maçonnées complexes en ayant recours à de la vision artificielle.

La construction sans cintre étant un enjeu important en ce qui concerne la productivité sur un chantier et la quantité de déchets produits, nous explorons, à cet effet, les possibilités qu’offre la rigidité en flexion inhérente aux maçonneries topologiquement autobloquantes. La génération de ces dernières, classique dans le cas plan, est généralisée ici à la conception de structures courbes, à partir de maillages de quadrangles plans et de manière paramétrique, grâce aux logiciels Rhinoceros 3D / Grasshopper. Pour cela, nous proposons un ensemble d’inégalités à respecter afin que la structure obtenue soit effectivement topologiquement autobloquante. Ces inégalités permettent, par ailleurs, d’introduire un résultat nouveau ; à savoir qu’il est possible d’avoir un assemblage de blocs dans lequel chacun des blocs est topologiquement bloqué en translation, mais un sous-ensemble – constitué de plusieurs de ces blocs – ne l’est pas. Un prototype de maçonnerie à topologie autobloquante est finalement conçu. Sa conception repose sur une découpe des joints d’inclinaison variable qui permet de le construire sans cintre.

En parallèle, nous abordons des aspects de vision artificielle robuste pour un environnement chantier, environnement complexe dans lequel les capteurs peuvent subir des chocs, être salis ou déplacés accidentellement. Le problème est d’estimer la position et l’orientation relative d’un bloc de maçonnerie par rapport à un bras robot, à partir de simples caméras ne nécessitant pas d’étape de calibration. Notre approche repose sur l’utilisation de réseaux de neurones convolutifs de classification, entraînés à partir de centaines de milliers d’images synthétiques de l’ensemble bras robot + bloc, présentant des variations aléatoires en terme de dimensions et positions du bloc, textures, éclairage, etc, et ce afin que le robot puisse apprendre à repérer le bloc sans trop de biais d’environnement. La génération de ces images est réalisée grâce à Unreal Engine 4. Cette méthode permet la localisation du bloc par rapport au robot avec une précision millimétrique, sans utiliser une seule image réelle pour la phase d’apprentissage ; ce qui constitue un avantage certain puisque l’acquisition de données représentatives pour l’apprentissage est un processus long et fastidieux. Nous avons également construit une base de données riche, constituée d’environ 12000 images réelles contenant un robot et un bloc précisément localisés, permettant d’évaluer quantitativement notre approche et de la rendre comparable aux approches alternatives. Un démonstrateur réel intégrant un bras ABB IRB 120, des blocs parallélépipédiques et trois webcams a été mis en place pour démontrer la faisabilité de la méthode.

Abstract

The context of this thesis work is the development of robotics in the construction industry. We explore the robotic construction of complex masonry structures with the help of computer vision.

Construction without the use of formwork is an important issue in relation to both productivity on a construction site and the amount of waste generated. To this end, we study topological interlocking masonries and the possibilities they present. The design of this kind of masonry is standard for planar structures. We generalize it to the design of curved structures in a parametrical way, using PQ meshes and the softwares Rhinoceros 3D and Grasshopper. To achieve this, we introduce a set of inequalities to respect in order to have a topological interlocked structure. These inequalities allow us to present a new result. Namely, it is possible to have an assembly of blocks in which each block is interlocked in translation, while having a subset – composed of several of these blocks – that is not interlocked. We also present a prototype of topological interlocking masonry. Its design is based on variable inclination joints, allowing construction without formwork.

In parallel, we are studying robust computer vision for unstructured environments like construction sites, in which sensors are vulnerable to dust or could be accidentally jostled. The goal is to estimate the relative pose (position + orientation) of a masonry block with respect to a robot, using only cheap cameras without the need for calibration. Our approach relies on a classification Convolutional Neural Network trained using hundreds of thousands of synthetically rendered scenes with a robot and a block, and randomized parameters such as block dimensions and poses, light, textures, etc, so that the robot can learn to locate the block without being influenced by the environment. The generation of these images is performed with Unreal Engine 4. This method allows us to estimate a block pose very accurately, with only millimetric errors, without using a single real image for training. This is a strong advantage since acquiring representative training data is a long and expensive process. We also built a new rich dataset of real robot images (about 12,000 images) with accurately localized blocks so that we can evaluate our approach and compare it to alternative approaches. A real demonstrator, including a ABB IRB 120 robot, cuboid blocks and three webcams was set up to prove the feasibility of the method.

Remerciements

Cette thèse n'aurait pas été possible sans l'aide et le soutien de nombreuses personnes que je tiens à remercier ici.

Je souhaite tout d'abord remercier les membres du jury qui ont accepté d'évaluer ma thèse, en particulier Antonella Mastrorilli pour avoir présidé ma soutenance et également Corentin Fivet et Vincent Lepetit qui ont accepté d'être rapporteurs. Leurs questions, avant et pendant la soutenance, ont permis d'avoir une discussion constructive.

Un grand merci à mes directeurs de thèse, Jean-François Caron et Renaud Marlet. Outre la confiance qu'ils m'ont accordée en me proposant ce sujet très intéressant et ambitieux, ils ont toujours été à l'écoute durant ces trois années. En me guidant sur les pistes les plus prometteuses, ils ont évité que je me disperse trop et ont ainsi joué un grand rôle dans l'accomplissement de cette thèse. Un grand merci également à Olivier Baverel et Mathieu Aubry qui m'ont encadré et fait profiter de leurs excellentes compétences dans leurs domaines respectifs.

Ce travail de recherche n'aurait pas été ce qu'il est sans le soutien de l'équipe technique. Ainsi, je tiens particulièrement à remercier Christophe Bernard pour son support sur le plateau numérique et Tristan Gobin pour toute l'aide et l'expérience qu'il m'a apportées pour l'utilisation des robots. Merci également à Gilles Moreau et Hocine Delmi pour leur aide. Merci à Marie-Françoise Kaspi pour son aide sur toutes les questions administratives.

Merci aux départements GCC et GMM de l'École des Ponts – notamment Alain Ehrlicher, Frédéric Tayeb et Laurent Maghdissian – pour les opportunités d'enseignement très enrichissantes qu'ils m'ont proposées. Un grand merci à Ioannis Stefanou qui, en plus de m'avoir fait confiance en me laissant enseigner Abaqus, m'a grandement aidé en partageant son expérience sur l'étude numérique du comportement des structures maçonnées.

Merci à Maurizio Brocato et Mathias Fantin pour m'avoir permis d'étudier expérimentalement les structures maçonnées aux Grands Ateliers de l'Isle d'Abeau et à la SNBR, ce qui m'a permis d'avoir une compréhension concrète de ce que j'étudiais théoriquement.

Merci à tous les professeurs, doctorants et stagiaires des équipes MSA et Imagine avec qui j'ai passé de très bons moments et qui ont rendu l'atmosphère de travail très agréable au quotidien. Un merci particulier à Philippe Hannequart et Pierre Margerit, deux doctorants et amis qui ont commencé leur thèse en même temps que moi et avec

qui l'on s'est soutenu et motivé tout au long de ces trois années.

Je souhaite remercier également tous mes amis, notamment ceux de la *coloc de Montreuil* sur qui j'ai toujours pu compter pour me changer les idées.

Enfin, merci à ma famille qui, même habitant loin, est toujours présente lorsque j'ai besoin d'elle et sur laquelle je peux toujours compter.

Table des matières

1	Introduction	1
1.1	Contexte	1
1.2	Robot Oriented Design pour la maçonnerie	3
1.2.1	Construction sans cintre pour un assemblage robotisé simplifié	3
1.2.2	Vision artificielle pour un environnement chantier	4
1.2.2.1	Objectifs et approche	4
1.2.2.2	Défis	6
1.2.2.3	Contributions	7
1.3	Organisation du manuscrit	8
I	État de l’art	11
2	Maçonnerie topologiquement autobloquante	13
2.1	Voûte plate d’Abeille	14
2.2	Assemblage topologiquement autobloquant	17
2.2.1	Définition	17
2.2.2	Exemples d’assemblages topologiquement autobloquants	18
2.3	Propriétés	19
2.4	Assemblage topologiquement autobloquant pour la construction	21
2.5	Introduction de courbure	23
2.6	Inclinaison variable des joints	23
2.7	Conclusion	24
3	Stabilité des structures maçonnées autobloquantes	25
3.1	Introduction	26
3.2	Modélisation des joints dans les maçonneries autobloquantes	26
3.2.1	Mécanismes au niveau des joints	27
3.2.2	Critères de résistance des joints	27
3.3	Polygone funiculaire et ligne de pression	30
3.4	Analyse limite pour les structures maçonnées	34
3.4.1	Analyse limite des maçonneries	34
3.4.2	Statique graphique	35

3.4.3	Equations d'équilibre discrétisées	36
3.4.4	Réseaux de forces et <i>Thrust Network Analysis</i>	36
3.4.5	Limitations de l'analyse limite et calcul à la rupture	37
3.4.6	Conclusion	39
3.5	Éléments finis continus	40
3.6	Méthode des éléments discrets	40
3.7	A propos des moteurs physiques type PhysX	42
3.8	Conclusion	42
4	Vision artificielle et estimation de pose	45
4.1	Apprentissage et réseaux de neurones	46
4.1.1	Apprentissage automatique et modèle	46
4.1.2	Apprentissage supervisé	47
4.1.2.1	Différents types d'apprentissages	47
4.1.2.2	Exemple : régression linéaire	48
4.1.3	Réseau de neurones artificiels	50
4.1.3.1	Neurone formel	50
4.1.3.2	Réseaux de neurones	51
4.1.3.3	Descente de gradient stochastique et époques	52
4.1.4	Surapprentissage	53
4.1.4.1	Exemple	54
4.1.4.2	Données d'entraînement et de validation	54
4.1.5	À retenir	56
4.2	Apprentissage à partir de données synthétiques	56
4.2.1	Adaptation de domaine	56
4.2.2	Simuler le réel	57
4.2.3	<i>Domain randomization</i>	57
4.3	Localisation d'un objet et estimation de pose	60
4.3.1	Base de données pour alignement 2D-3D	60
4.3.2	Méthodes basées sur l'alignement de modèle	60
4.3.3	Méthodes basées sur l'apprentissage	63
4.3.4	Position relative	64
4.4	Conclusion	65
II	Contributions	67
5	Assemblage topologiquement autobloquant sur surface courbe et condition de blocage	69
5.1	Méthode de construction d'assemblages topologiquement autobloquants à partir d'un maillage de quadrangles plans	70
5.1.1	<i>Moving cross-section procedure</i> et voûte plate d'Abeille	70
5.1.2	Notations et conventions d'orientation	72
5.1.2.1	Notations locales	72

5.1.2.2	Notations globales et compatibilité	74
5.1.3	Généralisation de la procédure à des maillages de surfaces courbes	75
5.1.4	Nécessité et utilité de tronquer	80
5.2	À propos du blocage	81
5.2.1	Conditions de blocage en translation	82
5.2.2	Cas particuliers des quadrangles plans	88
5.2.2.1	Quadrangle plan quelconque	88
5.2.2.2	Quadrangle parallèle	92
5.2.2.3	Applications	93
5.2.3	Blocage d'un sous-ensemble de bloc	97
5.2.3.1	Assemblage fortement topologiquement autobloquant	97
5.2.3.2	Conditions de blocage	98
5.2.4	Limites	100
5.3	Applications	100
5.3.1	Voûte à double courbure positive	100
5.3.1.1	Génération du maillage	100
5.3.1.2	Génération d'un assemblage topologiquement autobloquant	101
5.3.1.3	Calcul de stabilité	102
5.3.1.4	Résultats	105
5.3.2	Corolle	114
5.3.2.1	Analyse	116
5.3.2.2	Résultats	117
5.4	Conclusion du chapitre et ouverture	129
6	Localisation relative d'un bloc par rapport à un robot	131
6.1	Abstraction de la tâche	132
6.1.1	Description de la tâche	132
6.1.2	Robot et référentiel	135
6.2	Résolution de la tâche avec des CNNs et des images synthétiques	136
6.2.1	Estimation de pose relative avec un CNN	137
6.2.2	Base de données d'apprentissage synthétique	139
6.3	Résultats expérimentaux et analyse	142
6.3.1	Base de données d'évaluation	142
6.3.2	Métrique d'évaluation	145
6.3.3	Architecture du réseau	145
6.3.4	Apprentissage sur données synthétiques et validation sur données réelles	146
6.3.5	Agrégation des vues	147
6.3.6	Procédure en trois étapes et taux de succès de la saisie	147
6.3.7	Comparaison avec d'autres méthodes	150
6.3.8	Sensibilité à l'environnement	151
6.4	Conclusion	152

7	Conclusion et perspectives	155
7.1	Conclusion	155
7.2	Perspectives	157
A	Annexes	171
A.1	Base de données d'apprentissage composée d'images synthétiques	171
A.2	Base de données d'évaluation composée d'images réelles	172
A.2.1	Détails de l'architecture du réseau	174

Chapitre 1

Introduction

1.1 Contexte

Depuis les 20 dernières années, la croissance de la productivité dans le domaine de la construction est de seulement 1% par an alors qu'elle est de 3,6% pour l'industrie manufacturière. Pour les pays développés, tels la France ou les Etats-Unis, cette productivité est même stagnante (Barbosa et al., 2017). Ceci est notamment causé par une importante régulation de cette industrie, une forte dépendance au secteur public et des sous-investissements en recherche et développement. Parmi les sept domaines cités par Barbosa et al. (2017) pour améliorer cette productivité, on trouve entre autres un recours accru à l'automatisation et aux technologies digitales, notamment par l'accélération de la construction sur site grâce à la robotique. Pour accompagner cette révolution et développer les aspects de construction robotisée et de chantier du futur, des projets nationaux ont été lancés, notamment en Suisse (dfab¹) mais également en France (DiXite², Projet Matrice³).

La robotisation de la construction n'est en revanche pas une idée nouvelle. Bock and Langenberg (2014) nous rappellent que suite à la seconde guerre mondiale, des tentatives d'industrialisation du procédé de construction apparurent afin de faire face au défi de la reconstruction. Cela passa notamment par la préfabrication, la standardisation et la structuration du chantier – environnement généralement non structuré. Le choc pétrolier de 1973 a eu raison de cet élan en occident mais pas en Asie, notamment au Japon. En 1975, on assista au développement de robots de construction effectuant des tâches uniques, répétitives, de manière non autonome (Fig. 1.1a et 1.1b); l'utilisation de ce type de robots étant viable à la condition que le chantier soit restructuré à l'image d'une usine. Cela a mené à l'élaboration à partir de 1982 de *chantiers automatisés intégrés*, sorte d'usine mobile sur le site de construction dans laquelle des robots monotâches sont directement intégrés (Fig. 1.1c).

1. <http://www.dfab.ch/>

2. <https://dixite.future-isite.fr/>

3. <http://www.matrice-impression3d.fr/projet/>



(a) Shimizu Corporation, Robot de finition du béton, 1987



(b) T. Bock, ESPRIT 3 6450 ROCCO, 1992



(c) Obayashi Corporation, Concept de site de construction automatisé intégré, 1985

FIGURE 1.1 – Exemples de robots monotâches et de chantier automatisé intégré issus de Bock and Langenberg (2014)

À travers ces chantiers automatisés intégrés transparaît la nécessité de concevoir un chantier de manière spécifique pour une construction par des robots. C'est le concept de **Conception Orientée Robot**, ou bien **Robot Oriented Design** :

Robot-oriented design (ROD) is a central concept for enabling efficient construction automation and in particular integrated automated construction sites, all of which use, knowingly or unknowingly, ROD elements. It is concerned with the coadaptation of construction products and automated or robotic technology, so that the use of such technology becomes applicable, simpler, more robust, or more productive or efficient. (Bock and Linner, 2015)

Le ROD implique une co-adaptation. D'un côté, la conception de la construction est adaptée afin que cette dernière soit plus aisément réalisable par des robots ; d'un autre côté, les robots sont adaptés aux contraintes de la construction.

Dans cette thèse, nous explorons deux manières d'adopter une démarche ROD :

- en concevant différemment une structure maçonnerie afin qu'elle soit plus aisément constructible par des robots
- en développant une méthode de vision artificielle robuste à un environnement non structuré tel un chantier, utilisant des caméras non calibrées, afin d'apporter de la flexibilité aux robots

1.2 Robot Oriented Design pour la maçonnerie

Ce travail de thèse s'intéresse particulièrement à la construction robotisée de maçonneries – i.e. de constructions composées de blocs (briques, pierres, parpaing, etc) unis ou non par un liant. Cette problématique en tant que telle n'est pas nouvelle (cf. Fig. 1.1b), diverses équipes de recherche l'étudient (notamment Gramazio et Kolher à Zurich ou la Chair of Building Realisation and Robotics à Munich) et certaines entreprises proposent déjà des produits commerciaux pour empiler des briques avec des bras robots (Construction Robotics ou Fastbrick entre autres). Une des spécificités de ce travail est d'explorer une nouvelle manière de concevoir des maçonneries s'inspirant du ROD.

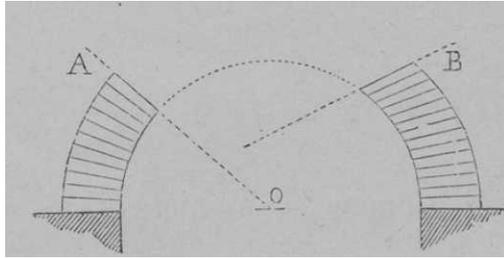
Premièrement, nous adaptons la forme des blocs pour pouvoir construire sans cintre, afin de rendre l'assemblage par des robots plus aisé. Deuxièmement, nous développons des algorithmes de vision artificielle, adaptée aux contraintes du chantier, pour localiser et estimer la pose (position + orientation) de ces blocs.

1.2.1 Construction sans cintre pour un assemblage robotisé simplifié

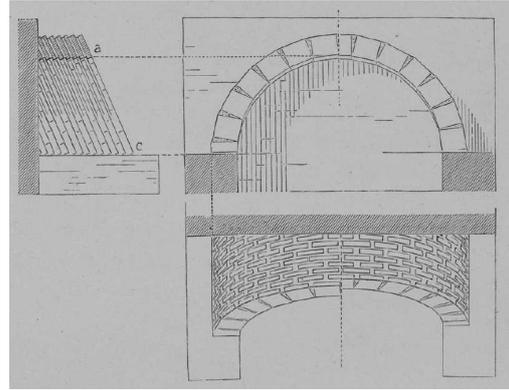
L'un des enjeux dans la construction d'ouvrages maçonnés est celui du franchissement. Comment construire au dessus d'un espace avec des éléments plus petits que l'espace à franchir? En général, une structure temporaire (un cintre, des étais, etc) est mise en place, puis enlevée une fois que toutes les pièces nécessaires à la stabilité sont installées. Pour un arc maçonné par exemple, il s'agit d'un cintre sur lequel sont placés les voussoirs de l'arc. Une fois le dernier bloc placé – la clé de voûte – l'arc est stable et le cintre peut être enlevé. Idem pour la construction en béton d'un plafond ou d'un plancher, le coffrage est enlevé une fois que le béton a pris. Néanmoins, en concevant la structure différemment et en changeant l'ordre d'assemblage, il peut être possible de limiter, voire d'éviter, le recours à ce type de structures temporaires.

Choisy (1883) expose par exemple les principes de construction sans cintrage des voûtes byzantines. Une manière classique de construire une voûte en berceau, mais nécessitant un cintre, est de la construire par "assises rayonnantes". Elle est construite rangée par rangée, en commençant par les rangées inférieures (Fig. 1.2a) puis une fois la dernière rangée supérieure placée (rangée de clés de voûte), le cintre peut être enlevé. Les Byzantins procèdent autrement : au lieu de construire rangée par rangée, ils construisent arcs par arcs. Une fois un arc construit, il est considéré stable et un autre arc peut être construit en s'appuyant sur l'arc précédent ; la cohésion du mortier étant suffisante pour maintenir les briques lorsque l'arc n'est pas complet. De plus, les arcs sont légèrement inclinés pour aider à la cohésion (Fig 1.2b). L'ordre d'assemblage ainsi qu'une conception différente (incliner les briques) permettent donc de faciliter une construction sans cintres. D'autres recherches plus récentes explorent également cette possibilité d'éviter ou limiter les cintres en jouant sur l'ordre d'assemblage (Deuss et al., 2014; Kao et al., 2017).

Cet enjeu n'est certes pas spécifique à la construction robotique, mais cette dernière peut apporter des réponses nouvelles. Nous explorons donc dans cette thèse une concep-



(a) En commençant par les rangées inférieures, la voûte surplombe progressivement le vide. Un cintre est nécessaire.



(b) En construisant arc par arc, en penchant les arcs et en utilisant l'adhésion du mortier, le cintre n'est pas nécessaire.

FIGURE 1.2 – Construction d'une voûte en berceau avec cintre et sans cintre selon Choisy (1883)

tion différente des structures maçonnées via une stéréotomie⁴ adaptée afin de construire de tels ouvrages sans cintre et d'éviter les coûts et difficultés inhérents à leur montage et à leur démontage. Pour cela, des maçonneries dites topologiquement autobloquantes sont étudiées et nous explorons les possibilités que confère la découpe particulière de leurs joints pour une construction sans cintre. Ainsi, l'assemblage par robot en est facilité ; la contrepartie étant que chaque bloc peut être différent et l'ordre d'assemblage non évident. Cette contrepartie n'est pas forcément un problème dès lors qu'on a recours au numérique, à des robots et à de la vision artificielle. En effet, une fois qu'un robot est entraîné à reconnaître les blocs et qu'il connaît les consignes d'assemblage, la variété des géométries des blocs et l'ordre d'assemblage non conventionnel ne pose pas forcément un problème supplémentaire. C'est par cette co-adaptation que la conception de ces maçonneries s'inscrit dans le ROD : les blocs sont conçus afin de simplifier l'assemblage par robot (car pas de cintres nécessaires) et le robot, grâce à la vision artificielle adaptée à l'environnement chantier, s'affranchit des contraintes liées à la multitude de géométries de blocs, rendant ainsi la construction plus efficace.

1.2.2 Vision artificielle pour un environnement chantier

1.2.2.1 Objectifs et approche

Un autre enjeu est de permettre aux robots d'opérer dans un environnement chantier, environnement peu structuré, en utilisant de simples caméras observant la scène. Là encore, cette problématique n'est pas nouvelle puisque permettre à un robot d'inter-

4. La stéréotomie est la science de la coupe des solides, terme essentiellement utilisé pour la coupe des pierres.

agir avec le monde en utilisant seulement de simples images est l'une des motivations initiales de la vision artificielle (Roberts, 1963). L'enjeu est dès lors de reconnaître les différents éléments composant la scène et leur position précise l'un relativement à l'autre. Plus concrètement, nous nous concentrons ici sur le problème de localiser et orienter des blocs de construction par rapport à un bras robot. Dans ce travail, nous poursuivons les deux objectifs suivants :

Robustesse aux changements d'apparence Contrairement à l'environnement relativement propre et structuré des usines, un chantier est un environnement hostile pour la vision artificielle. Les objets peuvent en effet changer d'apparences, à cause par exemple des variations de texture dues à la poussière et au mauvais temps, des occultations inopinées et des changements de luminosité dus aux conditions extérieures. Il y a donc une forte demande de robustesse en même temps qu'une grande précision exigée pour la manipulation. Afin d'accomplir la tâche proposée, une approche possible est d'aligner des modèles 3D correspondant au bras robot et aux blocs sur l'image, les localiser par rapport à la caméra et finalement les positionner les uns par rapport aux autres. Cependant, cela serait très sensible aux changements de forme et d'apparence, qui peuvent être omniprésents sur un chantier. Nous avons alors fait le choix de recourir uniquement à des méthodes d'apprentissage. Nous présentons ainsi une approche plus directe qui utilise les avancées récentes dans l'utilisation des réseaux de neurones convolutifs (CNNs). De tels réseaux ont l'avantage d'être génériques, ils peuvent être facilement entraînés pour être robuste à beaucoup de perturbations (notamment les occultations, les changements d'illumination ou de texture) et ils peuvent être appliqués aux images très efficacement.

Néanmoins, les CNNs requièrent une très grande quantité d'images d'entraînement annotées. Bien qu'il est possible d'animer automatiquement un robot et un bloc de construction devant une caméra pour créer une telle base de données, il faut généralement des milliers d'heures pour collecter des images mettant en scène des centaines de milliers de situations (Pinto and Gupta, 2016; Levine et al., 2018). De plus, une nouvelle base de données est nécessaire à chaque fois que le robot ou le bloc changent ; cette solution n'est pas praticable. Afin de la rendre générique et facilement applicable à d'autres robots et d'autres types de bloc, sans avoir besoin de la longue et fastidieuse élaboration d'une base de données d'images réelles pour l'apprentissage, nous avons fait le choix de recourir uniquement à des images synthétiques. Nous montrons alors une méthode flexible qui prend simplement en entrée des modèles 3D, sans texture, d'un bloc de construction (avec des variations paramétriques possibles) et le modèle 3D d'un robot (avec les positions possible de ses articulations). Cette méthode ne nécessite aucune image réelle. Elle est basée sur un apprentissage virtuel en utilisant seulement des images synthétiques et pourtant donne de bons résultats une fois évaluée sur des images réelles.

Utilisation de caméras non calibrées Dans des industries équipées de robots, effectuant des tâches répétitives et bien identifiées, des caméras pour le pilotage des robots peuvent être placées à des endroits pertinents et précis, et la détermination de leurs paramètres intrinsèques et extrinsèques – l’étape de calibration⁵ – peut être réalisée une fois pour toute. À présent, imaginons utiliser des robots sur un chantier. Puisque l’environnement est très peu structuré et puisque les tâches peuvent changer fréquemment, il n’y a pas de position de caméra générique permettant de voir toutes les actions du robot. Même si plusieurs caméras sont utilisées conjointement pour fournir assez de prises de vue pertinentes, certaines de ces caméras devront être souvent déplacées, soit manuellement (caméras sur trépieds), soit en utilisant des caméras montées sur des engins mobiles tels que des robots d’observations ou des drones. Dans ces conditions, on pourrait envisager de prendre le temps de recalibrer les caméras (vis-à-vis du robot) après chacun de leurs mouvements mais cela serait très contraignant. Nous visons donc un scénario de construction robotisée de maçonnerie qui ne nécessite pas d’étape de calibration car cela est difficilement compatible avec les conditions de travail sur un chantier. De plus, sur un chantier, les caméras peuvent subir des chocs ou être déplacées accidentellement, parfois involontairement. Une des options est d’accrocher directement les caméras sur les robots mais le nombre de caméras possibles ainsi que les points de vue accessibles sont drastiquement réduits. En outre, de telles caméras sont plus exposées à la poussière et aux accidents puisqu’elles sont plus proches des actions des robots. Nous avons alors considéré un scénario avec plusieurs caméras non calibrées – par exemple attachées à des trépieds que l’on bouge au fur et à mesure que la construction avance –, extérieures à la scène et qui l’observent, sans nécessairement de points de repère fiables qui permettraient un étalonnage absolu de la caméra. Un exemple d’une telle scène est représenté Figure 1.3.

1.2.2.2 Défis

Précision Les mors du préhenseur utilisé lors de nos expériences ont une course de 5 mm. La localisation du bloc doit donc être assez précise pour que le robot puisse effectivement saisir ce dernier. Le premier défi est donc de réussir à prédire précisément la position relative d’un bloc de maçonnerie par rapport à un bras robot, sans données de calibration et uniquement avec des techniques d’apprentissage.

Robustesse aux changements d’apparences Nous souhaitons estimer la pose (par rapport à un robot) d’objets de la catégorie composée de blocs parallélépipédiques aux dimensions variables. Dans les images réelles, les objets de cette catégorie peuvent subir des changements d’apparence dus à la poussière, à la saleté, aux variations de textures,

5. La calibration (ou étalonnage) des caméras est une étape permettant de trouver la relation entre les coordonnées 3D d’un point de l’espace et ses coordonnées 2D dans l’image de la caméra. Elle nécessite la détermination de paramètres intrinsèques – internes à la caméra – et extrinsèques – concernant la position de la caméra vis-à-vis de son espace de travail et par rapport aux autres caméras.

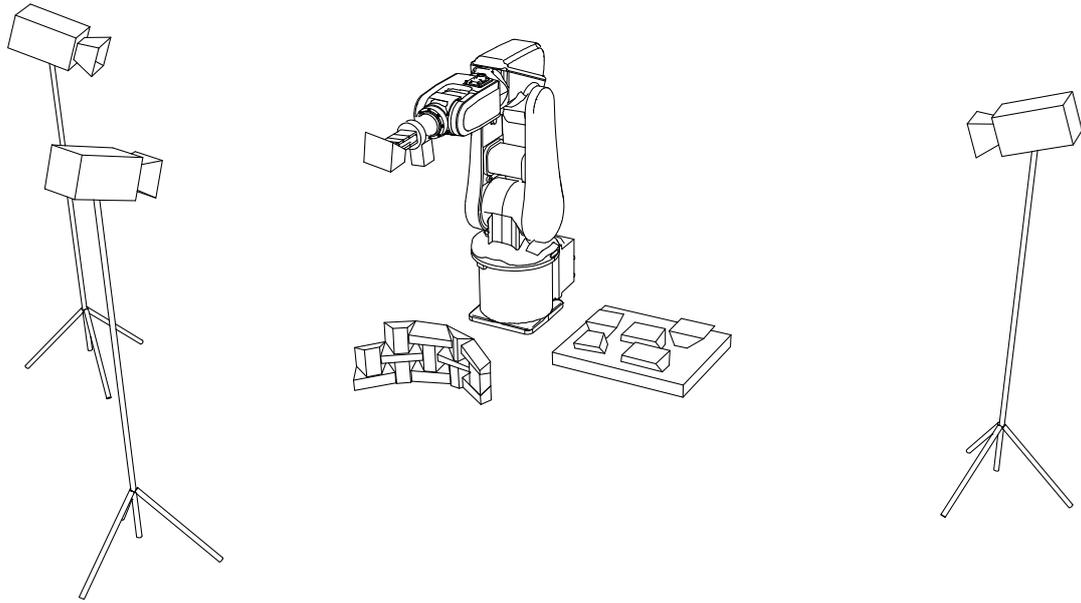


FIGURE 1.3 – Exemple de scénario de construction automatisée de maçonnerie sans calibration des caméras. Le robot peut éventuellement être mobile. Nous présentons dans le Chapitre 6 une méthode accomplissant l’estimation de pose relative pour une variante de ce scénario avec un seul bloc parallélépipédique.

d’illuminations, aux variations de l’arrière plan, etc . L’estimation de pose doit donc être invariante à ces changements d’apparences tout en étant suffisamment précise.

Écart de domaines entre images synthétiques et images réelles Nous souhaitons utiliser uniquement des images synthétiques pour l’apprentissage. Cependant, des images synthétiques et des images réelles peuvent être très différentes. Une méthode donnant de bons résultats sur des images synthétiques n’a a priori pas de raisons de donner de bons résultats sur des images réelles. Le dernier défi est donc de réussir à avoir de bons résultats sur des images réelles malgré un apprentissage sur des images synthétiques.

1.2.2.3 Contributions

Afin de répondre à cet enjeu de vision robuste dans un environnement chantier, nous explorons donc dans cette thèse la possibilité d’utiliser un dispositif ne nécessitant pas d’étape de calibration des caméras. Nos contributions sont les suivantes :

- Nous introduisons une nouvelle tâche robotique de localisation relative sans calibration des caméras. Nous la formulons en détail et fournissons une procédure

ainsi qu’une base de données d’images réelles pour son évaluation⁶. Notre approche inclue une procédure en trois étapes dans laquelle une première estimation de position grossière est réalisée, puis affinée une fois que le robot s’est déplacé vers la cible.

- Nous montrons que la localisation relative très précise (millimétrique) peut être atteinte avec uniquement des techniques d’apprentissage malgré le manque d’informations de calibration.
- Nous montrons que l’apprentissage peut être réalisé sans aucune image réelle, ouvrant la voie à de l’apprentissage générique avec des formes et des objets arbitraires.

Ce travail a donné lieu à une publication dans l’International Journal of Computer Vision :

Loing, V., Marlet, R., and Aubry, M. (2018). Virtual training for a real application : Accurate object-robot relative localization without calibration. International Journal of Computer Vision, 126 :1045-1060

1.3 Organisation du manuscrit

Cette thèse explore la conception de maçonnerie pour une construction sans cintre et une méthode de vision artificielle robuste adaptée à un environnement chantier, le tout afin de simplifier l’assemblage robotisé. Ce mémoire est organisé en deux parties.

La première partie expose l’état de l’art. Nous introduisons d’abord un type d’assemblages, les assemblages topologiquement autobloquants, principalement étudiés dans la communauté des matériaux mais utilisés parfois dans le domaine de la construction. De tels assemblages avec des joints d’inclinaison variable n’ont été que très peu étudiés (Chapitre 2) mais les utiliser en maçonnerie peut être une solution pour la construction sans cintre. Nous exposons ensuite les principales méthodes pour calculer la stabilité des structures maçonnées et montrons que celles-ci ne sont pas particulièrement adaptées à l’analyse de maçonneries dont la géométrie des joints est compliquée comme c’est le cas pour les assemblages topologiquement autobloquants (Chapitre 3). Cette première partie présente enfin l’apprentissage supervisé, les réseaux de neurones et différentes techniques de vision artificielle pour l’estimation de pose relative d’un bloc par rapport à un robot (Chapitre 4).

La seconde partie présente les contributions de cette thèse. Nous exposons d’abord une méthode de construction d’assemblages topologiquement autobloquants pour des surfaces courbes et avec des joints variables. Nous étudions plus précisément le phénomène d’autoblocage et montrons certaines particularités de ce dernier. Cette partie

6. La page du projet avec la base de données UnLoc (Uncalibrated Relative Localization) est <http://imagine.enpc.fr/~loingvi/unloc/>

expose notamment un prototype de maçonnerie constructible sans cintre, grâce à l'utilisation de joints variables (Chapitre 5). Enfin, nous montrons que l'apprentissage de réseaux de neurones convolutifs (CNN) sur des images synthétiques permet d'estimer la position relative d'un bloc par rapport à un bras robot dans des conditions proches d'un environnement chantier. Ces réseaux, entraînés sur images synthétiques mais utilisés sur des images réelles issues de simples caméras, permettent ainsi d'attraper un bloc parallélépipédique sans avoir besoin de calibrer les caméras (Chapitre 6).

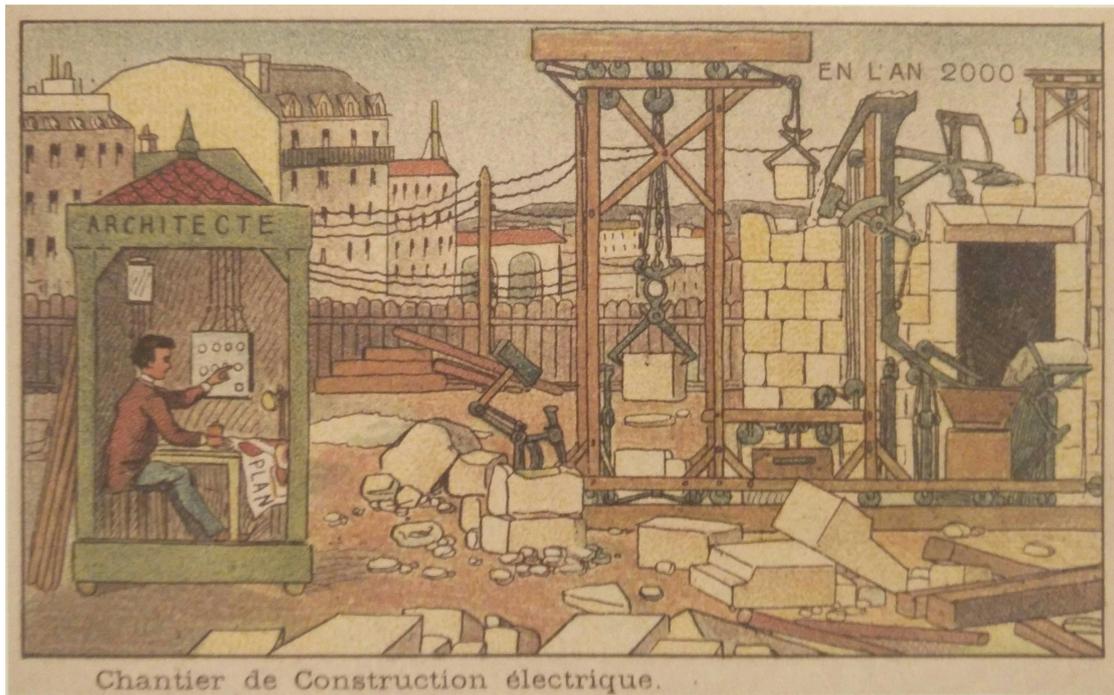


FIGURE 1.4 – Villemard (1910) Chantier de construction électrique, Visions de l'an 2000

Première partie
État de l'art

Chapitre 2

Maçonnerie topologiquement autobloquante

Résumé : Dans ce chapitre, nous présentons les maçonneries topologiquement autobloquantes par l'intermédiaire de la voûte plate d'Abeille. La géométrie particulière de ses blocs la rapproche des assemblages à topologie autobloquante. Nous présentons ces assemblages, certaines de leurs propriétés ainsi que leur utilisation dans le domaine de la construction. Nous montrons également que ce type de maçonneries avec des joints d'inclinaison variable ou issues de surfaces courbes complexes n'a été que très peu étudié.

2.1 Voûte plate d’Abeille

Généralement, la coupe des joints d’une structure clavée suit le *principe de normalité* (Fantin, 2017) qui implique la perpendicularité des joints avec l’intrados (Fig. 2.1).

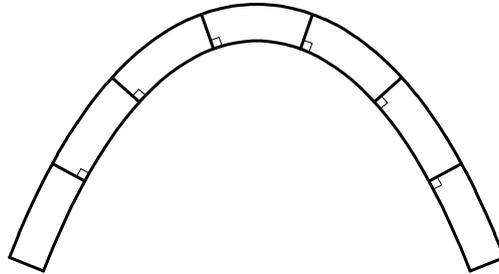
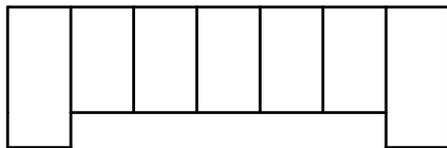
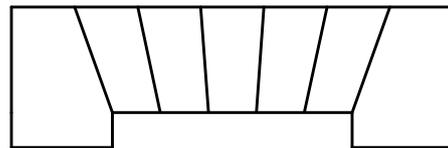


FIGURE 2.1 – Arc dont les joints sont perpendiculaires à l’intrados

Pour des raisons de stabilité, il n’est cependant pas toujours possible de suivre ce principe. C’est le cas notamment pour les plates-bandes ou les voûtes plates¹(Fig. 2.2). Pour ces structures, la coupe des joints a un rôle important dans la stabilité.



(a) Plate-bande à joints droits, perpendiculaires à l’intrados



(b) Plate-bande à joints inclinés

FIGURE 2.2 – Plates-bandes respectant ou non le principe de normalité. Pour avoir stabilité de la plate-bande à joints droits, la poussée aux appuis doit être très grande. Elle est moindre dans le cas des joints inclinés.

La voûte plate d’Abeille est un exemple de structure maçonnée ne suivant pas le principe de normalité. Inventée en 1699 par l’ingénieur architecte Joseph Abeille, améliorée quelques années plus tard par le père Sébastien Truchet (Fig 2.3), cette voûte a la particularité de permettre de couvrir une zone rectangulaire à partir d’éléments tous identiques et de taille inférieure à la zone à couvrir (Fig 2.4).

1. Une plate-bande est une structure semblable au linteau au dessus d’une ouverture dans un mur mais est composée de plusieurs blocs appelés *claveaux*. La voûte plate est une “voûte dont l’intrados est un plan horizontal” (Litré).

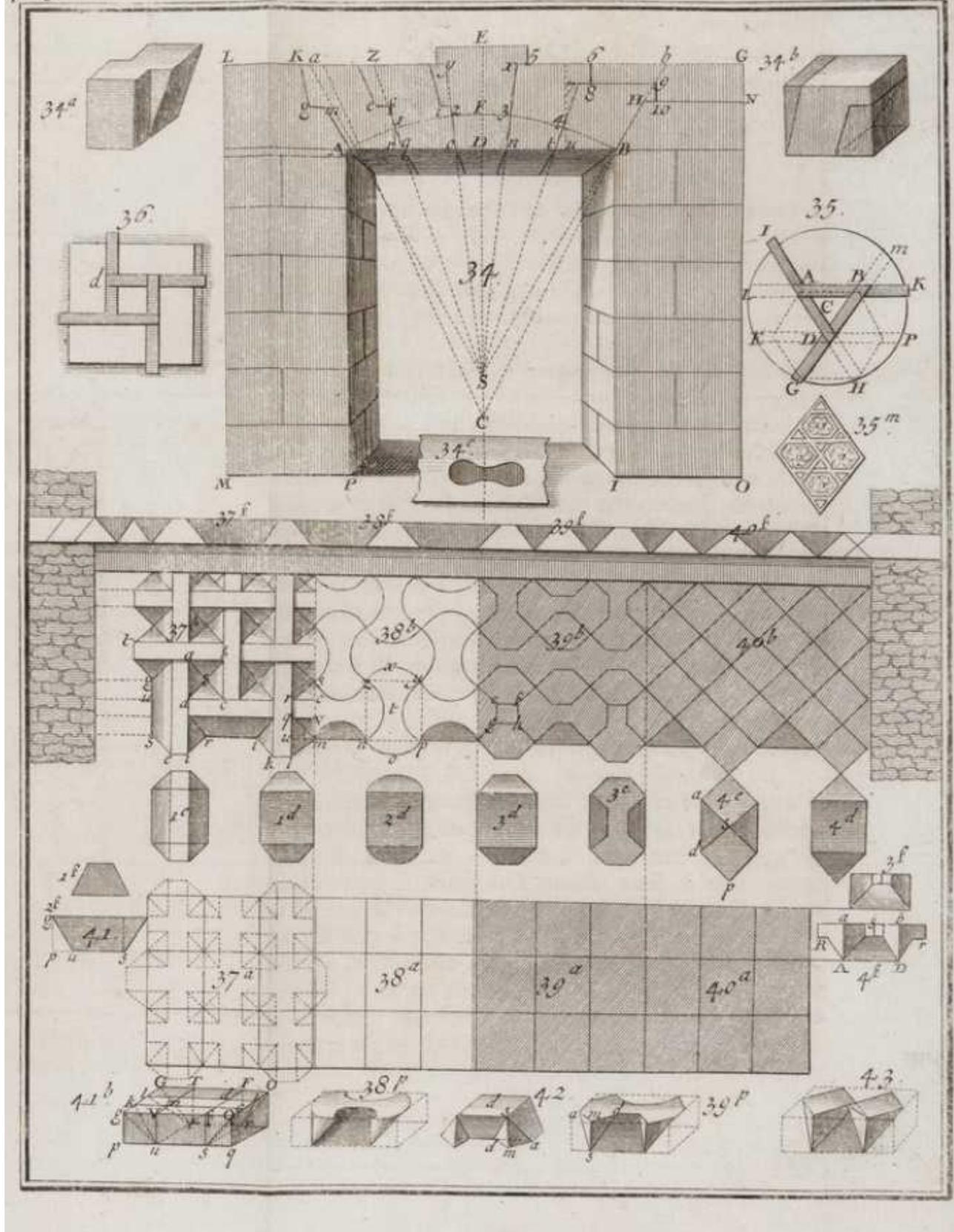


FIGURE 2.3 – Voûtes plates d’Abeille (37) et de Truchet (38) dans Frézier (1737). On remarquera aussi les structures réciproques (35) et (36).

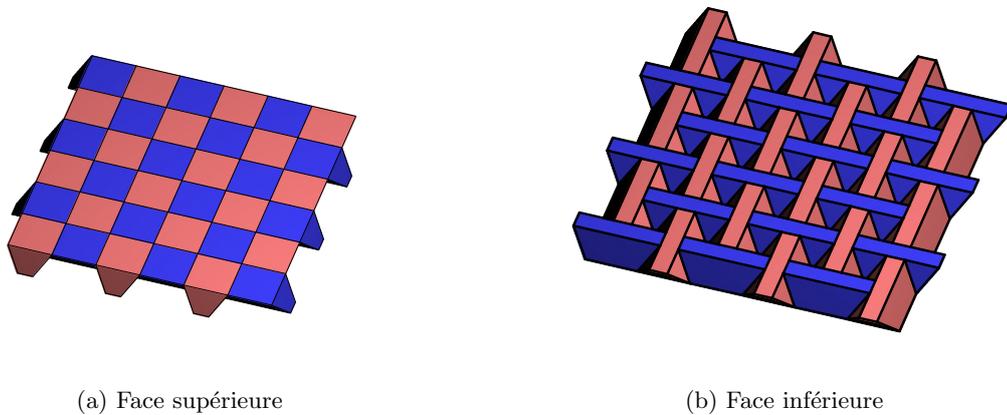


FIGURE 2.4 – Voûte plate d’Abeille.

Les claveaux² qui la composent sont des tétraèdres dont deux arêtes opposées sont tronquées (Fig 2.5). Une condition nécessaire à la stabilité d’une telle voûte plate est qu’elle soit contrainte à sa périphérie. Le principe sous-tendu par une telle structure est à rapprocher de celui des structures réciproques ; un bloc élémentaire est supporté par deux blocs voisins et supporte lui même deux autres blocs voisins. Bien que très astucieuse, cette invention n’a pas été très exploitée jusqu’à récemment. De par sa stéréotomie particulière, la voûte plate d’Abeille est un exemple d’**assemblage topologiquement autobloquant**.

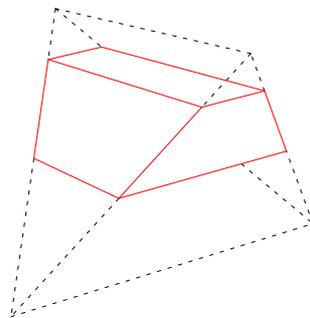


FIGURE 2.5 – Claveau d’une voûte plate d’Abeille. On remarque qu’il s’agit d’un tétraèdre dont les arêtes sont tronquées.

2. Un claveau est le bloc élémentaire d’une voûte plate ou d’une plate-bande. Un voussoir est celui d’un arc ou d’une voûte courbe. Dans la suite de ce manuscrit nous emploierons souvent par simplification le terme de *bloc*.

2.2 Assemblage topologiquement autobloquant

2.2.1 Définition

La voûte plate d'Abeille montre qu'il est possible de paver le plan par des tétraèdres (tronqués) (Fig. 2.6). Ces dernières années, ce type de pavage a été redécouvert au moins par deux fois. D'abord pour la conception de pavage pour la chaussée (Glickman, 1984) et, plus récemment, par la communauté des ingénieurs en matériaux (Dyskin et al., 2001a). En s'inspirant de la théorie du "bloc clé" dans la mécanique des roches – bloc qui empêche le mouvement de ses voisins dès lors qu'il est maintenu en place – Dyskin et al. (2001a) ont montré que dans un assemblage de tétraèdres réguliers disposés d'une certaine façon, il n'existe aucune cinématique pour sortir un élément de cet assemblage dès lors que les voisins sont fixes, comme expliquée Fig. 2.7. À condition que l'assemblage soit contraint à sa périphérie, les blocs sont bloqués sans avoir recours à des clés de blocage potentiellement sources de concentration de contrainte et plus difficiles à fabriquer. Puisque le blocage cinématique des blocs vient de la géométrie même de ces blocs et de la manière dont le plan a été divisé, ces types de matériaux ont été nommés **matériaux à topologie autobloquante** ou *topologically interlocked material* (TIM).

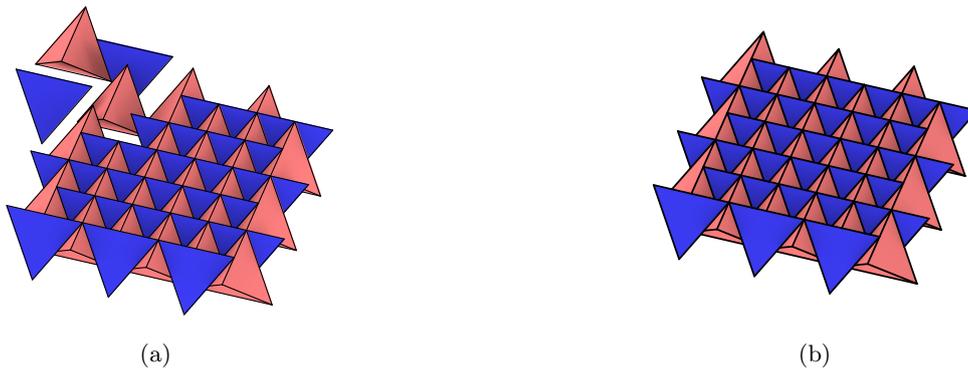


FIGURE 2.6 – Assemblage topologiquement autobloquant de tétraèdres réguliers.

Il n'y a pas de consensus entre les auteurs sur la définition d'autoblocage topologique. En effet, pour Ashby and Bréchet (2003), l'autoblocage topologique correspond à des structures discrètes et non liées capables de supporter des charges. Ainsi, un simple mur de briques serait un assemblage topologiquement autobloquant pour la compression (axiale) mais pas pour la traction. Néanmoins, l'usage semble consacrer le terme d'autoblocage topologique aux structures capables de supporter des charges selon toutes les directions et rotations. Notons cependant la différence entre le blocage topologique en translation, le blocage en rotation et le blocage topologique total. Dans la suite du manuscrit, en considérant les blocs de la périphérie de l'assemblage comme fixes, nous retiendrons donc les définitions suivantes (que l'on complétera dans le Chapitre 5) :

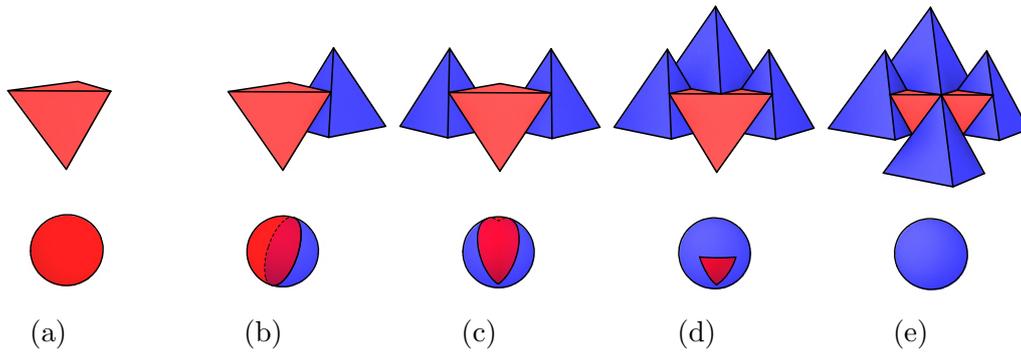


FIGURE 2.7 – **Blocage cinématique d’un tétraèdre.** Considérons les vecteurs pointant du centre de la sphère vers un point dans la zone rouge à la surface de cette sphère. Un mouvement de translation selon un de ces vecteurs permet d’enlever le tétraèdre considéré (a). Le contact plan-plan dû à l’ajout d’un tétraèdre voisin diminue cette zone rouge de moitié (b). L’ajout d’un deuxième (c), puis d’un troisième tétraèdre voisin (d) diminue davantage cette zone rouge, jusqu’à bloquer totalement le tétraèdre en translation (e).

Définition 1 : Dans un assemblage de blocs, un bloc est bloqué en translation (resp. rotation) s’il n’existe aucun mouvement infinitésimal de translation rectiligne (resp. rotation) possible permettant de l’enlever de l’assemblage en considérant ses voisins comme fixes.

Définition 2 : Un assemblage de N blocs est topologiquement autobloquant en translation (resp. rotation) si chacun de ses N blocs est bloqué en translation (resp. rotation).

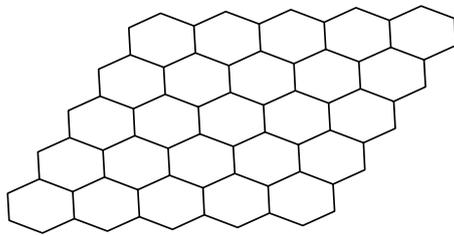
Définition 3 : Un assemblage de N blocs est **totalemment** topologiquement autobloquant s’il est topologiquement autobloquant en translation et en rotation.

Dyskin et al. (2003a) et Kanel-Belov et al. (2010) montrent l’équivalence entre blocage topologique en translation et blocage topologique total pour certains assemblages topologiquement autobloquants, en particulier ceux constitués de tétraèdres.

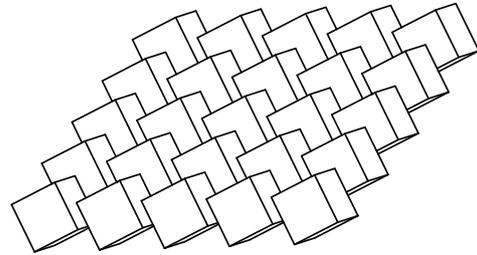
2.2.2 Exemples d’assemblages topologiquement autobloquants

Assemblages topologiquement autobloquants de solides de Platon Dyskin et al. (2001b, 2003b) proposent une méthode – plus tard nommée *moving cross-section procedure* (Kanel-Belov et al., 2010) – qui utilise des pavages réguliers du plan par des carrés, des cercles, des hexagones ou des décagones afin de construire des assemblages topologiquement autobloquants. Grâce à cette méthode, des assemblages constitués des

cinq solides de Platon³ sont ainsi obtenus (Kanel-Belov et al., 2010). Outre l'assemblage de tétraèdres de la Fig. 2.6, un exemple avec des cubes est présenté Fig. 2.8. Nous verrons plus en détail et développerons cette procédure au Chapitre 5.



(a) Pavage du plan par des hexagones réguliers



(b) Assemblage autobloquant de cubes

FIGURE 2.8 – Assemblage topologiquement autobloquant de cubes obtenu à partir d'un pavage du plan par des hexagones grâce à la procédure *moving cross-section*. Le cube est l'un des cinq solides de Platon.

Blocs ostéomorphes Les éléments des assemblages précédemment mentionnés sont tous des polyèdres convexes et ont donc des faces planes. Dyskin et al. (2003c) ont inventé un autre type d'assemblage topologiquement autobloquant, souvent rencontré dans la littérature, où le blocage se fait grâce la non-planéité des faces en contact. Vu la ressemblance de ces blocs avec des vertèbres, ils ont été qualifiés d'*ostéomorphes* (Fig. 2.9a). Un des avantages de tels assemblages sur ceux précédemment mentionnés est que l'intrados et l'extrados sont "lisses". De plus, en choisissant bien les paramètres géométriques, il est possible de faire des coins (Fig. 2.9b). Ces blocs ne sont plus convexes mais si la concavité est faible, on évite a priori de trop fortes concentrations de contraintes (Dyskin et al., 2003c).

2.3 Propriétés

Un assemblage topologiquement autobloquant peut être étudié d'un point de vue matériau ou bien d'un point de vue structure.

Vu comme un matériau, un tel assemblage se classe parmi les matériaux architecturés, plus précisément, architecturés dans le but de renforcer par la fragmentation (Ashby and Bréchet, 2003; Dyskin et al., 2001b). Les interfaces entre blocs empêchent en effet la propagation de fissures ; une ruine locale n'entraînant pas une ruine globale, contrairement à un matériau monolithique. Molotnikov et al. (2007) a d'ailleurs montré que les assemblages de blocs ostéomorphes ont un palier de percolation de 25%. C'est-à-dire que sur un assemblage de dimension infinie, on peut en moyenne retirer aléatoirement 25%

3. Les cinq solides de Platon sont le tétraèdre, le cube, l'octaèdre, le dodécaèdre et l'icosaèdre.

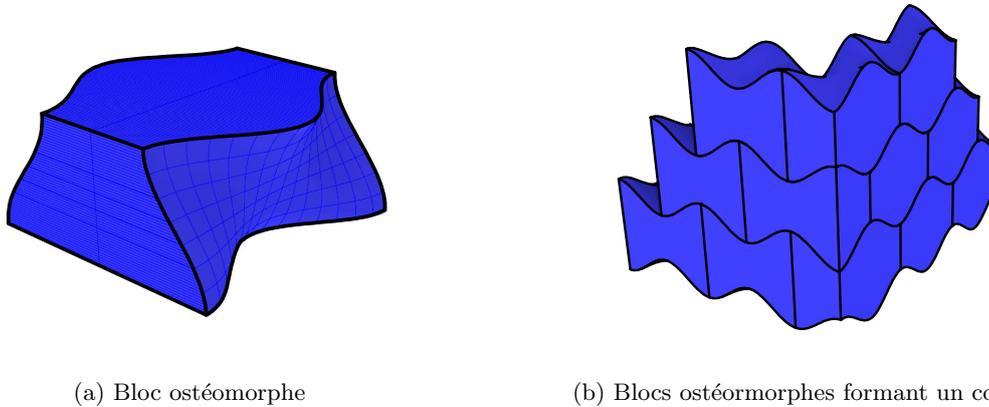


FIGURE 2.9 – Bloc ostéomorphe et assemblage de blocs ostéomorphes formant un coin.

des blocs sans ruine de l'assemblage. Cette propriété n'est néanmoins pas vraie pour les assemblages de cubes.

Les principaux essais effectués sur de tels assemblages sont des essais d'indentations. L'assemblage est maintenu dans un cadre rigide permettant une précontrainte latérale, le plus souvent variable. Une poussée verticale est exercée vers le bas en son centre (voir Fig 2.10). Les valeurs d'efforts et de déplacements sont ensuite récupérées pour étudier le comportement matériau. Ces essais ont été effectués pour des assemblages de tétraèdres (Dyskin et al., 2001a; Khandelwal et al., 2012), de cubes (Estrin et al., 2004; Dugué, 2013) ou de blocs ostéomorphes (Dyskin et al., 2003c; Autruffe et al., 2007; Brugger et al., 2008; Dugué, 2013). Le comportement est similaire pour ces différents types de blocs : lors de la phase de chargement, les courbes force-déplacement font ressortir une phase linéaire élastique suivi d'un comportement ductile. À la fin de la décharge, il y a présence d'un déplacement résiduel. Par rapport à une plaque monolithique équivalente, la rigidité initiale est plus faible. Plus la pré-compression latérale est grande, plus cette rigidité est grande (voir Fig 2.11). À noter qu'avec des blocs dans un matériau fragile, l'assemblage garde ce même comportement ductile (Khandelwal et al., 2012; Dugué, 2013). Autruffe et al. (2007) a étudié la sensibilité au coefficient de frottement en effectuant des tests sur des blocs ostéomorphes en glace. En augmentant la température de la glace, le coefficient de frottement diminue. Il a ainsi été montré que plus le frottement est faible, plus la rigidité initiale au chargement est faible, et l'énergie dissipée pour une charge maximum équivalente, quant à elle, est plus grande ; la flèche résiduelle étant elle même plus grande.

Vu comme une structure, les propriétés étudiées ne sont pas exactement les mêmes. Tout d'abord, l'échelle considérée est de l'ordre du centimètre, voire du mètre, une échelle plus typique du milieu de la construction. De plus, on s'intéresse plutôt à la stabilité



FIGURE 2.10 – Essai d’indentation d’un assemblage de cubes en aluminium (issu de Estrin et al. (2004))

de la structure sous certains chargements, à la transmission des efforts entre blocs, à la descente de charge ou à la poussée aux bords. De ce point de vue, la littérature est moins fournie.

Fleury (2009) a effectué un essai de chargement d’une voûte plate d’Abeille d’une portée de l’ordre de 2 m. Il en ressort que le ratio poussée/charge est de l’ordre de 0.5 pour des angles de joints de 60° avec l’horizontal, et donc comparativement plus faible que celui d’une plate-bande de dimensions similaires (cf. Fig. 2.12). Plus les joints deviennent verticaux, plus la poussée est grande. À l’inverse, plus l’angle des joints avec l’horizontal diminue, moins la poussée est grande ; mais les voussoirs travaillent davantage en flexion, augmentant le risque de fissure du matériau (ici de la pierre) peu résistant en traction. Brocato and Mondardini (2015) ont aussi étudié le comportement structurel de la voûte plate d’Abeille. Il montre qu’elle se comporte plus comme une grille structurelle fléchie que comme une voûte en compression. En considérant les claveaux comme des poutres épaisses, ils montrent que l’énergie de déformation est principalement une énergie de déformation de flexion.

2.4 Assemblage topologiquement autobloquant pour la construction

Yong (2011) envisage plusieurs applications pour les assemblages topologiquement autobloquant en génie civil : les murs de soutènement, la résistance au séisme et le pavage de chaussée. Nous pensons par ailleurs que les assemblages autobloquants peuvent aussi être utilisés en tant que mur pare-blocs du fait de la ductilité apportée par l’autoblocage, de la résistance à la percolation et en se basant sur les travaux de Feng et al. (2015) sur la réponse mécanique aux impacts des assemblages topologiquement autobloquants. Récemment, de tels assemblages ont déjà été utilisés dans la construction, principalement sous forme de prototype. Par exemple pour un mur préfabriqué, un dôme ou bien une arche (Brocato et al., 2014; Brocato and Mondardini, 2010; Fallacara, 2016). Notons que

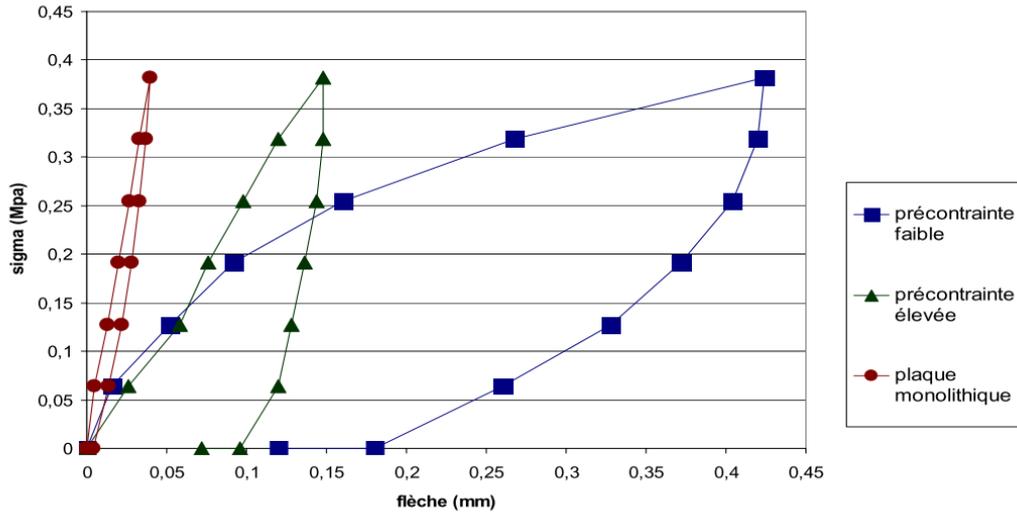
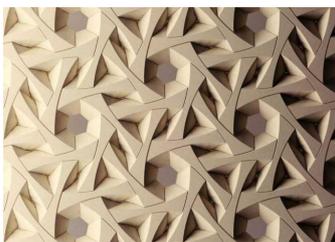


FIGURE 2.11 – Influence de la précontrainte sur l’indentation d’un assemblage de blocs ostéomorphes en glace (issu de Brugger et al. (2008))

Text proportions	Abeille	1 way platband	2 way platband	Drawing proportions	Abeille	1 way platband	2 way platband
Thrust (in N)	12 580	31 936	13 699		8 909	31 858	13 283
Weight (in N)	26 212	27 922	27 922		16 135	18 212	18 212
Ratio	0.48	1.14	0.49		0.55	1.75	0.73

FIGURE 2.12 – Rapport poussée/charge pour une voûte Abeille d’une portée de l’ordre de 2 m et comparaison avec des plates-bandes classiques (issu de Fleury (2009))

la forme des claveaux est légèrement différente (Fig 2.13a) ou bien que de la courbure a été introduite pour aller plus loin qu’une simple voûte plate (Fig 2.13b).



(a) "Modular stone wall" par G. Fallacara



(b) "Stone Matters" par AAU Anastas.



(c) Voûte en berceau de type Abeille par G. Fallacara à la biennale de Venise.

FIGURE 2.13 – Maçonneries courbes topologiquement autobloquantes

2.5 Introduction de courbure

Plusieurs approches sont utilisées pour introduire de la courbure dans de tels assemblages. Fallacara (2006) part d'une forme simple (en l'occurrence une voûte plate Abeille ou de Truchet) à laquelle il impose une déformation topologique paramétrique grâce à des modificateurs, comme des modificateurs de courbure ou des modificateurs de torsion. Il effectue des transformations sur ces voûtes plates à l'aide de ces modificateurs pour obtenir des voûtes en berceau (Fig 2.13c). Il part du principe que c'est plus aisé de partir d'un modèle simple et de le déformer plutôt que définir la forme globale (courbe) et faire la modélisation 3D des claveaux directement en prenant en compte l'autoblochage. Brocato and Mondardini (2010) génèrent un assemblage de type Abeille en forme de dôme en utilisant une méthode utilisée pour la génération des nexorades (structures réciproques). Tessmann (2012) explore la possibilité d'assemblage autobloquant de tétraèdres dans le domaine courbe mais en facetant les tétraèdres, perdant de ce fait la propriété de convexité du bloc et de planéité des faces, empêchant la possibilité d'avoir une fabrication par découpe plane.

Enfin, les dernières approches utilisées pour générer des assemblages topologiquement autobloquants courbes sont des approches bottom-up s'inspirant de la procédure *moving cross-section* de Kanel-Belov et al. (2010) ; cette méthode est expliquée en détail au Chapitre 5. Weizmann et al. (2016) en utilisent une variante pour construire un assemblage de tétraèdres à partir d'une surface courbe. Néanmoins, les blocs obtenus ont des faces non planes. Vella and Kotnik (2016) s'en sont également inspirés en utilisant des maillages par des quadrangles de surfaces courbes, pour obtenir des assemblages courbes de tétraèdres non réguliers mais avec des faces planes. Cette méthode permet également de créer des ouvertures dans l'assemblage. C'est une méthode similaire à celle – trouvée indépendamment – utilisée dans ce travail de thèse. Elle est exposée plus en détail dans le Chapitre 5.

2.6 Inclinaison variable des joints

Les maçonneries topologiquement autobloquantes sur surface courbe avec des joints d'inclinaison variables ont été peu étudiées. Brocato and Mondardini (2012) changent les angles pour faire de l'optimisation structurelle. Vella and Kotnik (2016) génèrent des assemblages sur surfaces courbes mais en utilisant le même angle (par rapport à la surface neutre) à chaque fois. Le changement d'inclinaison des angles dans les assemblages topologiquement autobloquants n'a en fait pas vraiment été abordé. Nous pensons que pouvoir changer pour chaque bloc l'inclinaison des joints peut apporter des paramètres de conception intéressants. En effet, en considérant de tels assemblages comme un matériau architecturé, en se basant sur une même géométrie globale, on peut générer des microstructures différentes (Fig. 2.14). L'optimisation de cette microstructure pouvant éventuellement apporter des effets bénéfiques dans la stabilité, la résistance au séismes, etc. Dans cette thèse, nous nous intéressons particulièrement à la possibilité qu'offrent des

joints d'inclinaison variable pour l'assemblage des maçonneries sans cintre. Par ailleurs, en utilisant des joints d'inclinaison variable, la propriété d'autoblocage peut disparaître, comme nous le verrons dans le Chapitre 5.

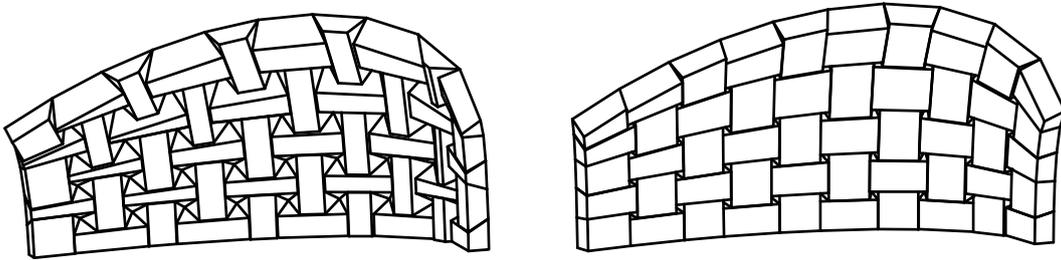


FIGURE 2.14 – Deux assemblages autobloquants obtenus à partir d’une même surface courbe mais avec des microstructures différentes.

2.7 Conclusion

Dans ce chapitre, les assemblages topologiquement autobloquants ainsi que certaines de leurs propriétés ont été présentés. Nous avons également évoqué la manière de concevoir ces assemblages sur des surfaces courbes. Utilisé dans le domaine de la construction, ce type de maçonnerie autobloquante ne respecte pas le principe de normalité pour la coupe des joints (cf. Sec. 2.1). Cette coupe de joints particulière confère à la structure à la fois une certaine rigidité en flexion et un comportement ductile – même si le matériau est fragile. Nous verrons au Chapitre 5 que nous utilisons ces propriétés en définissant des inclinaisons de joint variables pour construire sans cintre.

Chapitre 3

Stabilité des structures maçonnées autobloquantes

Résumé : Le chapitre précédent présente les assemblages topologiquement autobloquants, certaines de leurs propriétés mécaniques et aborde différentes méthodes pour les générer. Dès lors que l'on envisage d'utiliser ce type d'assemblage dans le domaine de la construction ou en génie civil, il est intéressant de connaître leur comportement sous l'action d'un chargement imposé. Ce chargement pouvant être un simple chargement de poids propre ou bien un chargement extérieur dissymétrique (par exemple l'action du vent). On se limite ici aux chargements statiques. Le présent chapitre présente de manière non exhaustive les principales approches possibles pour l'analyse de la stabilité des structures maçonnées en mettant en avant les avantages et inconvénients de chacune dans le cas particuliers des structures topologiquement autobloquantes. Nous montrons que mises à part les techniques par éléments finis ou discrets coûteuses en temps, les techniques habituelles d'analyse de stabilité ne sont pas adaptées pour les maçonneries autobloquantes courbes avec joints d'inclinaison variable.

3.1 Introduction

Lorsque l'on s'intéresse à la stabilité d'une structure sous un chargement donné, plusieurs questions peuvent légitimement se poser. Un chargement imposé est-il supporté par la structure et est-il proche du chargement ultime? Autrement dit, quel est le coefficient de sécurité pour un tel chargement? Pour un motif de chargement donné, quel est le chargement maximal menant à la ruine de la structure?

En ce qui concerne spécifiquement les structures maçonnées, la ruine n'est pas due à la ruine du matériau constitutif de ses blocs mais plutôt à une instabilité liée à la géométrie des blocs et à la distribution de la masse (Heyman, 1995; O'Dwyer, 1999). L'exemple de O'Dwyer (1999) est parlant pour illustrer ces propos. Il considère une force ponctuelle appliquée sur un arc qui provoquerait la ruine de celui-ci alors que l'intensité de cette force est bien inférieure à la contrainte à la rupture du matériau (Fig. 3.1). En première approximation, il est donc courant de considérer que la ruine des structures maçonnées est liée à une rupture de l'équilibre et non pas à des considérations matériaux. Cette remarque justifie l'usage de modèles réduits pour l'analyse de stabilité. Si un modèle réduit de la structure est stable, en passant à l'échelle, la stabilité sera conservée.

Ce chapitre expose les techniques d'analyse utilisées classiquement pour répondre à ces questions et met en avant les avantages et inconvénients de ces techniques pour les maçonneries topologiquement autobloquantes.

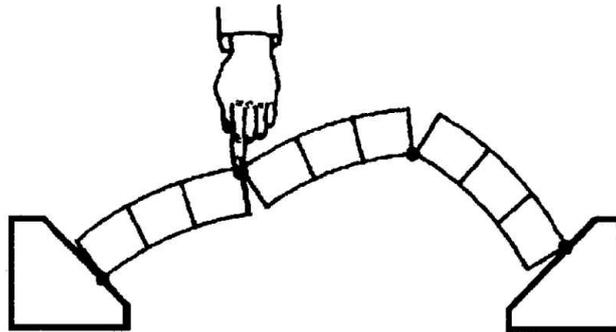


FIGURE 3.1 – Ruine d'un modèle d'arc composé de voussoirs rigides ; la charge de ruine est indépendante de la résistance des voussoirs (O'Dwyer, 1999)

3.2 Modélisation des joints dans les maçonneries autobloquantes

Une conséquence directe de cette remarque introductive est que l'étude de la résistance des structures maçonnées s'effectue plutôt en s'intéressant à la résistance de ses joints. C'est principalement l'apparition de mécanismes au niveau des joints qui va mener à la ruine de la structure maçonnée, et non la résistance du matériau constitutif.

3.2.1 Mécanismes au niveau des joints

Les interfaces entre deux blocs peuvent être remplies de plâtre, de mortier ou bien n'utiliser aucun mortier (on parle alors de joint sec). Pour les constructions anciennes, en raison de la dégradation du mortier, le comportement des joints est associé à un comportement de joints secs. De même, les structures autobloquantes que nous modéliserons présenteront des joints secs. En présence de mortier, nous négligerons l'épaisseur de celui-ci. Nous considérerons donc par la suite le joint comme une surface plane sans épaisseur.

Nous pouvons dès lors séparer les mécanismes de rupture au niveau des joints en deux groupes, heymanien et non-heymanien (Fig. 3.2), ces adjectifs faisant référence au théorème d'Heyman expliqué en Section 3.4.1. Dans les mécanismes heymaniens, la composante tangentielle du déplacement est nulle. Inversement, dans les mécanismes non-heymaniens, la composante tangentielle est non nulle (Bagi, 2014).

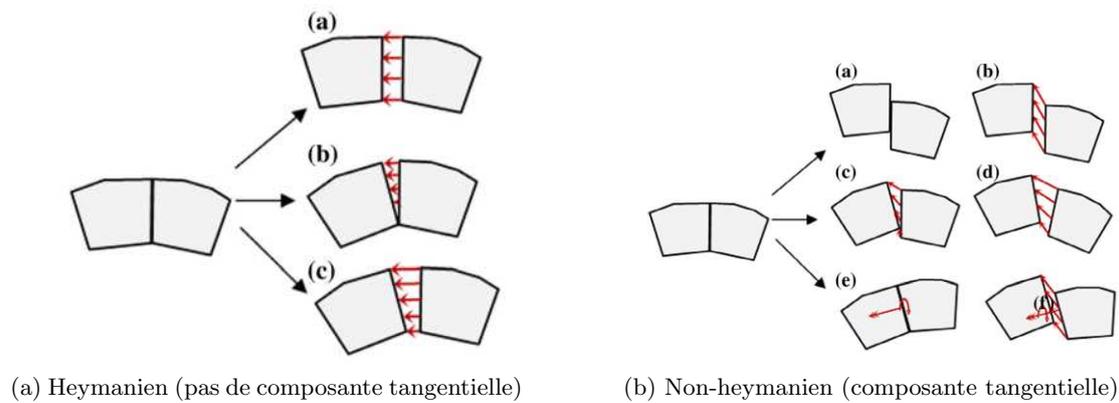


FIGURE 3.2 – Déplacement au niveau des interfaces de contact (Bagi, 2014)

3.2.2 Critères de résistance des joints

Pour modéliser le contact aux interfaces, il faut à la fois modéliser le critère de résistance de l'interface et sa règle d'écoulement – c'est-à-dire la cinématique de rupture de l'interface lorsque la limite du critère est atteinte. Plusieurs modélisations du contacts aux interfaces sont possibles. Citons par exemple, les interfaces lisses sans frottement, les interfaces à frottement de type Tresca (Fig. 3.3a), les interfaces à frottements de type Coulomb (Fig. 3.3b) et les interfaces parfaitement rugueuses.

Critère de Tresca Selon le critère de Tresca (Fig. 3.3a), pour un état de compression non nulle ($F_n < 0$), le glissement peut se produire si la composante tangentielle de l'effort au niveau de l'interface F_t atteint la valeur limite k . Pour un état de compression nulle ($F_n = 0$), le décollement peut se produire. En notant, u_n et u_t les composantes normales et tangentielles de la discontinuité de vitesse au niveau de l'interface, les cinématiques

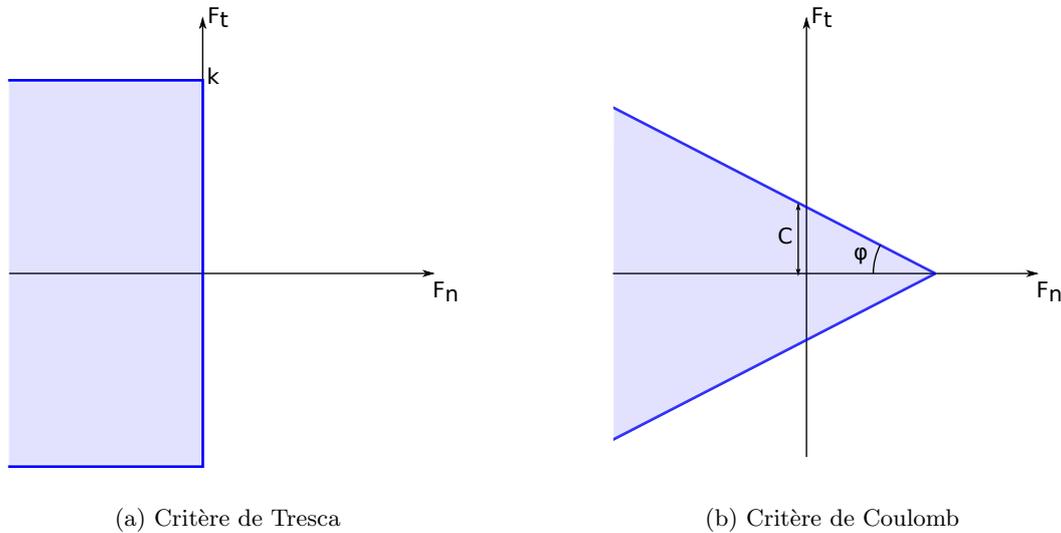


FIGURE 3.3 – Exemples de critères de résistance aux interfaces. F_n et F_t sont respectivement les composantes normales et tangentielles de l’effort au niveau de l’interface. $F_n < 0$ correspond à de la compression.

associées (par la règle de normalité¹) à ce critère sont :

- si $F_n < 0$ et $|F_t| = k \Rightarrow u_n = 0$ et $u_t \neq 0$ (glissement)
- si $F_n = 0$ et $|F_t| < k \Rightarrow u_n \geq 0$ et $u_t = 0$ (décollement
sans composante tangentielle)
- si $F_n = 0$ et $|F_t| = k \Rightarrow u_n \geq 0$ et $u_t \neq 0$ (décollement
avec composante tangentielle)

Les interfaces lisses sans frottement et les interfaces parfaitement rugueuses sont des cas particuliers du critère de Tresca, avec respectivement $k = 0$ et $k = \infty$. Pour les interfaces parfaitement rugueuses, la limite du critère est atteinte seulement si F_n est nulle, et les cinématiques associées sont telles que $u_n \geq 0$ et $u_t = 0$; elles n’ont pas de composantes tangentielles. On remarque que les cinématiques heymaniennes de la Fig. 3.2 sont associées au critère de résistance des interfaces parfaitement rugueuses.

Pour le critère de Tresca (et ses deux cas particuliers) et les cinématiques associées, la règle de normalité est vérifiée.

Critère de Coulomb Le critère de Coulomb (Fig. 3.3b) exprime la dépendance de l’effort tangent F_t par rapport à l’effort normal F_n appliqué au niveau de l’interface à

1. La règle de normalité est à différencier du principe de normalité introduit en Sec. 2.1. Le principe de normalité correspond à une découpe des joints perpendiculairement à l’intrados. Alors qu’ici, la règle de normalité réfère à la modélisation du comportement plastique; à savoir qu’en plasticité associée, le taux de déformation plastique est dirigée suivant la normale extérieure au domaine d’élasticité. Pour de plus amples informations, le lecteur pourra se référer à Salençon (2002).

travers le domaine de résistance défini par :

$$f = |F_t| + F_n \tan(\phi) - C \leq 0 \quad (3.1)$$

où ϕ est l'angle de frottement et C la cohésion. La valeur maximum de la composante normale de l'effort en tension est parfois limitée :

$$F_n \leq F_{max} \quad (3.2)$$

L'angle de frottement dépend de l'état des interfaces considérées et peut changer de valeur du fait de la dégradation de celle-ci, notamment lors d'un essai de cisaillement cyclique (Lourenço and Ramos, 2004). Pour un angle de frottement non nul, la cinématique associée à ce critère a une composante normale non nulle ($u_n > 0$). C'est-à-dire que même pour un état de compression non nul ($F_n < 0$), lorsque la limite du critère est atteinte, il y a décollement des blocs. Le frottement est dit **dilatant**. Cependant ce phénomène ne correspond pas à une réalité physique. En effet, pour une précompression modérée, les joints en maçonnerie suivent en général un critère de Coulomb (Papadopoulos et al., 1998; Pluijm, 1999; Lourenço and Ramos, 2004). Or pour des joints secs, la cohésion est considérée nulle ($C = 0$) et les expériences montrent que l'angle de dilatation – qui représente le déplacement normal à la surface accompagnant le glissement – est généralement très faible et considéré égal à 0 (Papadopoulos et al., 1998; Lourenço and Ramos, 2004). Autrement dit, il n'y a pas décollement des blocs lorsqu'ils glissent l'un par rapport à l'autre. L'aspect dilatant des joints de Coulomb ne représente donc pas bien le comportement réel des joints en maçonnerie. Pour déterminer les cinématiques d'écoulement du critère on introduit alors un potentiel d'écoulement plastique de la forme :

$$g = |F_t| + F_n \tan(\psi) - c \leq 0 \quad (3.3)$$

$$F_n \leq F_{max} \quad (3.4)$$

où ψ est l'angle de dilatation.

Les cinématiques du critère sont alors associées à ce potentiel g et non plus à f . Si l'angle de frottement et l'angle de dilatation sont égaux, $\phi = \psi$, le critère et l'écoulement sont associés, la règle de normalité est respectée; on parlera de critère de Coulomb associé. Dans le cas contraire ($\phi \neq \psi$), la règle de normalité n'est pas respectée et on parlera de critère de Coulomb non associé.

La valeur de l'angle de dilatation joue un rôle important dans la résistance des structures maçonnées confinées (Godio et al., 2018). Des joints dilatants surestiment le multiplicateur de rupture (coefficient de sécurité) et cette surestimation est d'autant plus significative pour la maçonnerie confinée, ce qui est le cas pour les voûtes plates d'Abeille, et les assemblages topologiquement autobloquants.

Nous retiendrons donc de cette section, qu'en première approche, pour une maçonnerie topologiquement autobloquante à joints secs, le critère à considérer est celui de type

Coulomb non associé avec un angle de dilatation nul. Ce critère ne respecte pas la règle de normalité.

3.3 Polygone funiculaire et ligne de pression

Afin d'introduire certaines notions et d'exposer différentes problématiques liées au calcul de stabilité des structures maçonnées, intéressons nous, en tant qu'exemple, à l'analyse de stabilité d'un arc sous un chargement donné. Considérons seulement le poids propre et modélisons cet arc comme plan et constitué de n blocs rigides (Fig. 3.4a). Dans cet exemple, chaque bloc est soumis à trois forces : le poids et la réaction des deux voisins (Fig. 3.4b). Le respect de l'équilibre des forces impose au polygone des forces d'être fermé. Le respect de l'équilibre en moment impose aux lignes d'action de ces forces d'être concourantes en un point. En écrivant les équations d'équilibre en force et en moment pour chaque bloc, on obtient un système de $3 * n$ équations indépendantes et $3 * (n + 1)$ inconnues correspondant aux composantes des forces résultantes aux joints ainsi qu'à leur point d'application. La structure est donc hyperstatique d'ordre 3. Imposer la valeur de trois inconnues, par exemple le point d'application et les deux composantes d'une réaction aux appuis, permet de résoudre ce système.

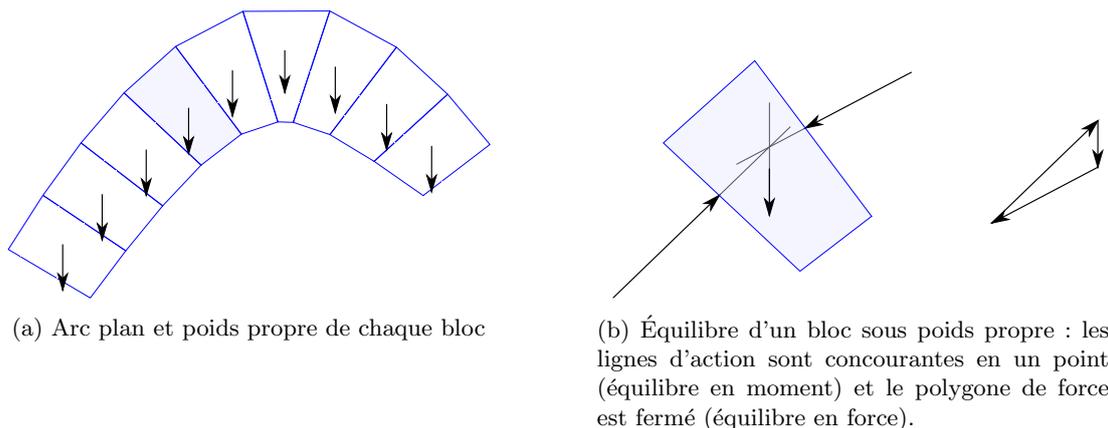


FIGURE 3.4 – Arc soumis à son poids propre

Polygone funiculaire En traçant l'ensemble des polygones de forces de chaque bloc, on obtient le **polygone funiculaire** (Fig. 3.5a). Graphiquement, le choix de deux des trois inconnues hyperstatiques revient au choix de la position du pôle noté O . La troisième inconnue hyperstatique est choisie, par exemple, à travers le choix du point d'application d'une des réactions aux appuis. Une convention couramment utilisée est de tracer le polygone funiculaire directement sur l'arc en reportant les lignes d'actions des forces s'exerçant sur chaque bloc (Fig. 3.5b). Changer la valeur des inconnues hyperstatiques permet de tracer différents polygones funiculaires (Fig. 3.6). Les deux polygones funiculaires de la Fig. 3.6 représentent chacun un système de forces en équilibre avec le poids

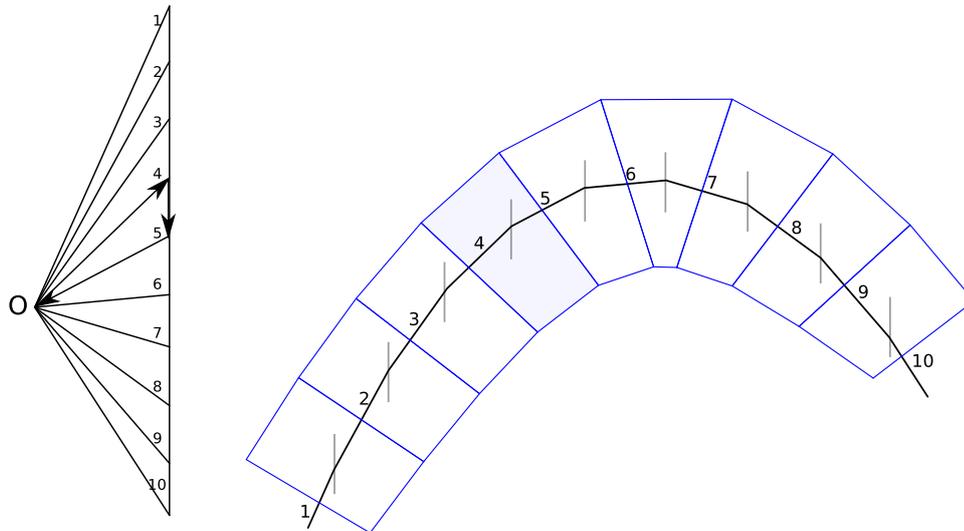
propre. A priori, sans autre informations, il n'est pas possible de savoir lequel de ces systèmes de forces en équilibre s'applique réellement. Il n'est également pas possible de conclure sur la stabilité de l'arc sous poids propre. Le théorème d'Heyman présenté en Sec. 3.4 donne des réponses à ces questions.

Ligne de pression Pour un bloc donné, le point d'application d'une force est l'intersection du joint avec la droite d'action de la force s'appliquant sur celui-ci. L'ensemble des points d'application des forces résultantes donne une suite de points nommée **ligne de pression**. Bien qu'il s'agisse d'une suite de points, c'est le terme de *ligne* qui est couramment utilisé. Dans la figure Fig. 3.6b les lignes de pressions sont représentées par les points rouges et bleus.

Un polygone funiculaire d'un arc ne dépend pas des joints; seule la donnée du chargement est nécessaire pour le tracer, non celle des joints. En revanche, par définition, la ligne de pression dépend des joints. Comme rappelé dans Moseley (1843); Rankine (1858); Fantin and Ciblac (2016); Fantin (2017), ces deux concepts – polygone funiculaire et ligne de pression – sont souvent très proches graphiquement et parfois confondus. Mais dans certains cas, les deux lignes ne sont pas confondues (voir Fig. 3.7). Ce phénomène est davantage expliqué dans la figure Fig. 3.8.

De plus, selon l'époque et les auteurs, diverses appellations ont été utilisées, prêtant parfois à confusion. Retenons qu'en anglais la *ligne de pression* s'est appelée *line of resistance* et que l'on parle désormais de *thrustline*. Le lecteur peut se référer à Fantin (2017) pour plus de précision.

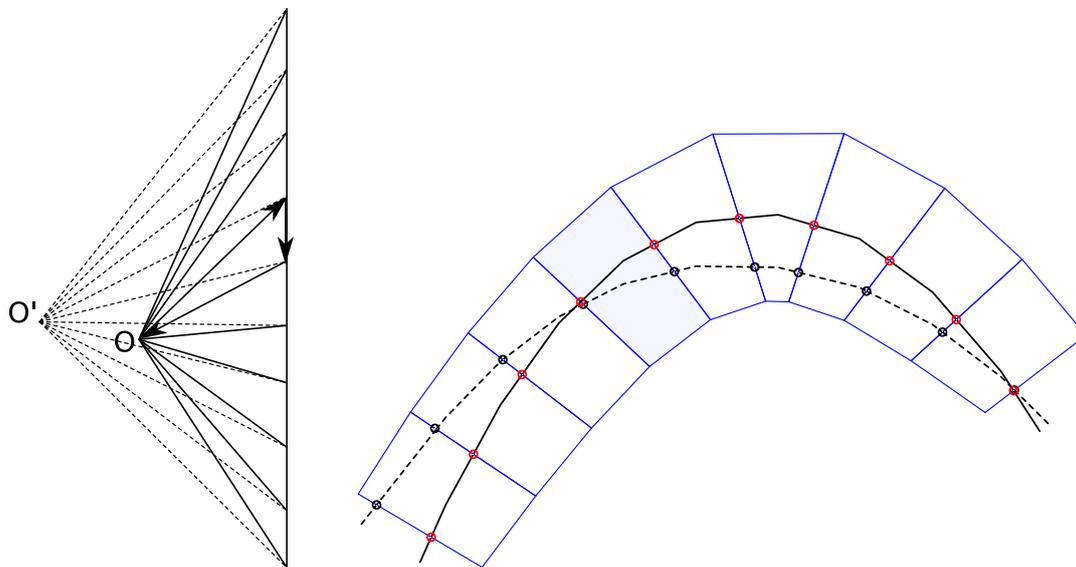
Ces notions de ligne de pression et de polygone funiculaire vont nous permettre d'introduire le calcul à la rupture des structures maçonnées, qui constitue l'une des principales approches d'analyse de stabilité des structures maçonnées. Nous expliciterons par la suite d'autres approches possibles plus adaptées à notre étude, telles que le calcul aux éléments finis ou bien le calcul aux éléments discrets.



(a) Polygone funiculaire

(b) Polygone funiculaire tracé sur l'arc. Les lignes verticales représentent les lignes d'actions du poids de chaque bloc. Les lignes notées 1 à 10 correspondent aux lignes d'actions des forces du polygone funiculaire (a).

FIGURE 3.5



(a) Polygones funiculaires avec 2 pôles différents O et O'

(b) Polygones funiculaires tracés sur l'arc pour différentes valeurs d'inconnues hyperstatiques. Les intersections des droites d'actions avec les joints correspondants – la ligne de pression – sont également tracées (points rouges et bleus).

FIGURE 3.6 – Représentations de deux systèmes de forces en équilibre avec le poids.

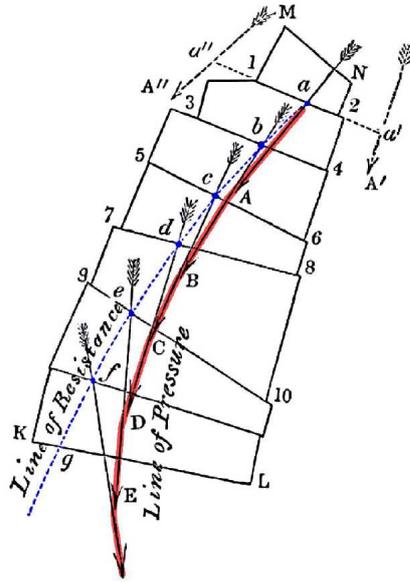
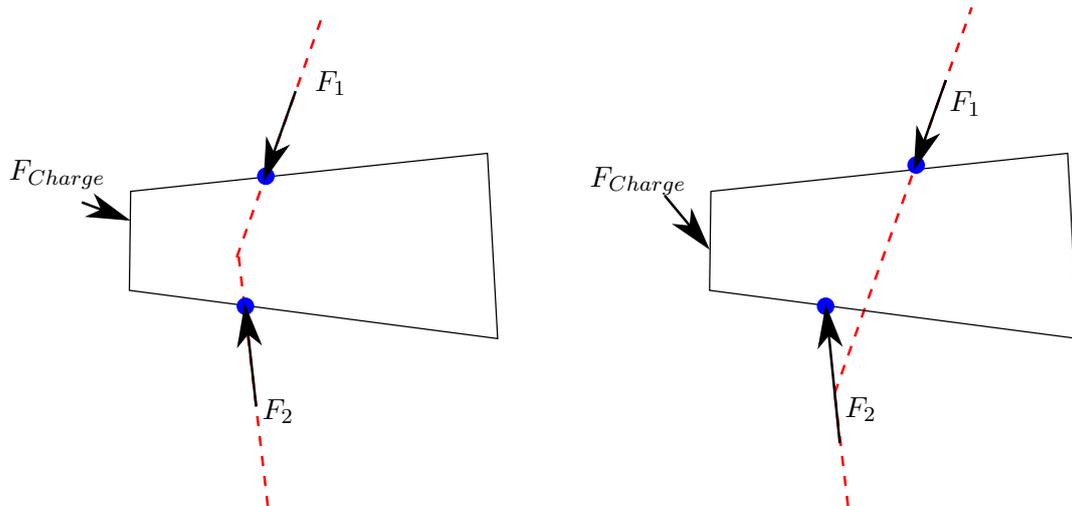


FIGURE 3.7 – Ligne de pression en bleu (line of resistance) et polygone funiculaire en rouge (line of pressure), d'après Moseley. Ici, les deux lignes ne sont pas confondues. Issue de Fantin (2017)



(a) Les droites d'action des forces s'exerçant aux interfaces s'intersectent dans le corps du bloc. Le polygone funiculaire et la ligne de pression sont confondus.

(b) Les droites d'action des forces s'exerçant aux interfaces ne s'intersectent pas dans le corps du bloc. Le polygone funiculaire et la ligne de pression ne sont pas confondus.

FIGURE 3.8 – Selon le chargement F_{Charge} et la disposition des forces F_1 et F_2 s'appliquant aux interfaces d'un bloc, le polygone funiculaire (en pointillé rouge) et la ligne de pression (points bleus) ne sont pas nécessairement confondus.

3.4 Analyse limite pour les structures maçonnées

Dans le cas général, l'analyse limite s'intéresse à l'ensemble des chargements supportés par un système constitué d'un matériau élastique et parfaitement plastique obéissant au principe du travail plastique maximal (Salençon, 1983). Dans la modélisation des milieux continus, pour un chargement donné, s'il existe un champ de contraintes respectant à la fois les équations d'équilibre, les conditions aux limites et le critère de résistance de ce matériau parfaitement plastique, alors le chargement est supporté par le système. Dans le cas d'une modélisation par solides rigides, s'il existe un ensemble d'efforts (forces et moments) satisfaisant les équations d'équilibre et les conditions aux limites – un tel ensemble d'efforts étant qualifié de **statiquement admissible** – et respectant le critère de résistance du matériau parfaitement plastique obéissant au principe du travail plastique maximal, alors le chargement est supporté par le système. Notons que dans une modélisation par solides rigides, le critère de résistance du matériau doit être entendu comme étant un critère s'appliquant sur les interfaces de contact entre solides et considérant la loi de frottement associée (cf. Section 3.2.2). Puisque les critères de résistance des interfaces sont convexes, il suffit de vérifier que la règle de normalité est respectée.

3.4.1 Analyse limite des maçonneries

Heyman (1982) a généralisé l'analyse limite aux structures maçonnées. Il considère, bien que de manière implicite, les blocs comme infiniment rigides et introduit les trois hypothèses suivantes pour le matériau :

1. aucune résistance en traction
2. résistance en compression infinie
3. pas de glissement des blocs entre eux

La première hypothèse correspond au fait qu'un joint entre deux blocs n'est pas capable de transmettre des efforts de traction. La deuxième hypothèse renvoie à la remarque introductive (Section 3.1) selon laquelle les efforts considérés sont faibles comparés à la résistance du matériau. La troisième hypothèse permet, en première approximation, au critère de résistance des joints de respecter la règle de normalité (et ainsi de pouvoir appliquer les conclusions de l'analyse limite). Comme vu en Section 3.2.2, ce n'est pas exactement vrai ; le non glissement des blocs n'impliquant pas le respect de la règle de normalité. Nous verrons en Section 3.4.5 une correction à cette hypothèse.

En s'appuyant sur ces hypothèses, Heyman adapte l'approche par l'intérieur de l'analyse limite, *thrustline* correspondant à la *ligne de pression* :

However, a very powerful statement can be made by translating the 'safe' (or lower-bound) theorem of plasticity into terms applicable to masonry. It

is this : If a thrust line can be found, for the complete arch, which is in equilibrium with the external loading (including self weight), and which lies everywhere within the masonry of the arch ring, then the arch is safe. The importance of this theorem lies in the fact that the thrust line found in this way need not be the actual thrust line. To demonstrate that the arch will stand as a structure it is necessary to show only that there is at least one satisfactory internal force system; viewed anthropomorphically, the arch is at least as clever as the analyst, and it will discover for itself an equally satisfactory position for the pole of the force polygon. (Heyman, 1982)

L'analyse limite appliquée à l'exemple de l'arc consiste donc à trouver, dans le mode de chargement considéré, une ligne de pression entièrement incluse dans l'épaisseur de l'arc pour en assurer sa stabilité. La force de cette méthode est que cette ligne de pression, qui dépend des inconnues hyperstatiques choisies pour la tracer, représente un système d'efforts en équilibre avec le chargement, mais il s'agit d'un système parmi d'autres. Il ne représente pas forcément l'état réel du système, mais trouver une telle ligne de pression dans l'épaisseur de l'arc certifie que le chargement sera effectivement supporté. De plus, aucune information n'est nécessaire sur l'histoire du chargement, sur les contraintes initiales, ni sur les déformations.

Le travail de l'ingénieur consiste dès lors à trouver un système de forces en équilibre avec le chargement considéré. Si l'ensemble des points d'application de ces forces (la ligne de pression dans le cas de l'arc 2D) se trouve bien au niveau des interfaces de contact de chaque joint, alors le chargement est supporté. En général, on cherche à construire l'ensemble des chargements supportés et à en estimer les chargements limites afin d'établir un facteur de sécurité. Pour approximer l'ensemble des chargements supportés et estimer les chargements limites, les différentes approches classiquement utilisées sont l'approche statique par l'intérieur, l'approche statique par l'extérieur et l'approche cinématique par l'extérieur. On ne s'attardera pas davantage sur ces deux dernières approches, le lecteur pouvant se référer à Salençon (1983, 2002) pour plus de précision. L'approche statique par l'intérieur consiste à paramétrer le chargement et trouver des efforts statiquement admissibles dans ce mode de chargement tout en respectant le critère de résistance du matériau. C'est cette approche qui est utilisée dans le théorème d'Heyman.

Les paragraphes suivants exposent différentes méthodes pour trouver des ensembles d'efforts statiquement admissibles dans le cas particulier des structures maçonnées.

3.4.2 Statique graphique

La statique graphique du XIX^e siècle (Culmann, 1875) permet de trouver un ensemble d'efforts statiquement admissibles dans le mode de chargement étudié. Dans le cas 2D, cette méthode est assez pratique pour tracer les polygones funiculaires puis les lignes de pression si la stéréotomie est connue. C'est cette approche qui a été utilisée implicitement pour tracer les polygones funiculaires des figures Fig. 3.5 et Fig. 3.6. Avant la disponibilité des outils numériques, la statique graphique étant difficilement praticable en 3D, l'étude

de stabilité des structures simples 3D se réalisait en se ramenant à un problème 2D. Par exemple, pour s’assurer de la stabilité d’une coupole, on coupait celle-ci en tranche le long des méridiens. La stabilité de chacune de ces tranches assimilées à des arcs impliquait alors la stabilité de la coupole. Cette méthode ne prend cependant pas en compte les effets entre les tranches, notamment l’effet des cercles horizontaux de compression ou de traction. Ces dernières années, les outils numériques ont néanmoins permis de développer la statique graphique pour le cas 3D (Schrems and Kotnik, 2013; Akbarzadeh et al., 2015; D’Acunto et al., 2016).

3.4.3 Equations d’équilibre discrétisées

En se basant sur Charnes and Greenberg (1951), Livesley (1978) transforme la recherche d’efforts statiquement admissibles avec le chargement respectant les critères de résistance du matériau en un problème d’optimisation linéaire qui peut être résolu de manière algorithmique par ordinateur, en discrétisant les équations d’équilibre pour chaque bloc. Il l’applique d’abord au problème de l’arc 2D puis dans les cas 3D (Livesley, 1992). Il insiste néanmoins sur le fait que lorsque le mécanisme de ruine est dû au glissement et que le frottement est de type Coulomb non associé, le facteur de charge est sur-évalué. Cette approche a été plus récemment utilisée par Smars (2000) et améliorée par Whiting (2011) pour la conception de structures maçonnées saines en apportant une mesure de l’infaisabilité.

3.4.4 Réseaux de forces et *Thrust Network Analysis*

Pour généraliser la notion de polygone funiculaire au cas 3D, O’Dwyer (1999) a développé la méthode des réseaux de forces, qui s’applique pour des chargements discrets et verticaux. Un réseau de forces consiste en un maillage dont chaque nœud se situe sur la ligne d’action des chargement verticaux. Chaque branche du maillage correspond à la direction des forces de compression s’appliquant sur un nœud. Les coordonnées des nœuds dans le plan horizontal étant fixées, les paramètres de la méthode sont les valeurs des composantes horizontales des forces et la hauteur des nœuds du maillage. Un processus numérique itératif sur ces paramètres permet ensuite au réseau de forces de respecter les équations d’équilibre. Le *Thrust Network Analysis (TNA)* (Block et al., 2006; Block, 2009) améliore cette méthode par l’introduction de deux diagrammes réciproques, le diagramme de forme et celui de force. Ce TNA, initialement développé pour l’analyse de structures existantes, est particulièrement adapté pour la recherche de forme des structures funiculaires et mis en œuvre dans RhinoVAULT (Rippmann, 2016). Nous ne détaillerons pas plus ces méthodes ici mais nous noterons néanmoins qu’elles sont une extension au 3D du polygone funiculaire et non de la ligne de pression ; la donnée des joints n’est en effet pas nécessaire, ni prise en compte. Néanmoins, si un tel réseau est compris dans l’épaisseur de la maçonnerie, le théorème de l’analyse limite permet encore une fois de conclure sur la stabilité et le calcul du facteur de sécurité de la structure.

Nous venons d’exposer trois types de méthodes pour trouver des ensembles d’efforts

statiquement admissibles afin de pouvoir appliquer les résultats de l'analyse limite : la statique graphique, la discrétisation des équations d'équilibre et les réseaux de forces. Nous pouvons aussi citer la résolution analytique possible pour des géométries simples.

3.4.5 Limitations de l'analyse limite et calcul à la rupture

L'analyse limite semble donc être adaptée pour l'étude de stabilité des maçonneries topologiquement autobloquantes. Cependant, certaines limitations que nous allons exposer font que ce n'est pas nécessairement le cas, car nous accordons notamment une certaine importance à la prise en compte des joints des structures topologiquement autobloquantes.

Domaines des solutions d'équilibre L'analyse limite pour la maçonnerie selon Heyman consiste à trouver un système de forces en équilibre avec le chargement (i.e. statiquement admissible) ; si les points d'application de ces forces sont au niveau des interfaces de contact (et non en dehors), alors le chargement est supporté. Le TNA que l'on vient de mentionner est une méthode pour trouver un système de forces en équilibre avec le chargement. Cependant, cette méthode considère une extension du polygone funiculaire et non de la ligne de pression. Or il peut arriver que les deux soient différents (cf. Section 3.3). En cherchant à ce que cette extension du polygone funiculaire soit dans l'épaisseur de la maçonnerie, on se limite à un domaine des solutions d'équilibre plus restreint. Il peut y avoir des cas, pour des stéréotomies particulières, dans lesquels le polygone de force est en dehors de l'épaisseur de la maçonnerie mais les points d'application des forces sont bien au niveau des interfaces.

De plus, avec le TNA, l'espace de recherche des solutions d'équilibre est également limité par le fait que cette méthode ne se préoccupe que d'équilibre avec des forces concourantes en un point. Or, il est possible d'avoir un équilibre en force et moment avec des forces non concourantes (Fig. 3.9). (Fantin, 2017) souligne que cet espace de recherche restreint est trop limité pour trouver des solutions stables pour les voûtes plates d'Abeille avec angle de frottement faible (cf. Fig. 3.10). Fantin and Ciblac (2016) ont étendu la méthode des réseaux de forces par l'ajout de branches partielles additionnelles, méthode que nous ne détaillerons pas ici. Cette méthode étendue a été appliquée avec succès aux voûtes plates, notamment celle d'Abeille (Fantin, 2017), mais elle ne s'applique pas aux voûtes d'Abeille non planes – et plus généralement aux maçonneries topologiquement autobloquantes non planes. Une piste envisageable pour étendre cette méthode à des voûtes de type Abeille non planes est de trouver une transformation conservant l'équilibre statique et permettant de passer d'une voûte plate à la voûte courbe voulue. De telles méthodes de transformation sont notamment présentées dans Fivet (2016).

Frottement non associé et calcul à la rupture Pour appliquer l'analyse limite, le matériau doit être parfaitement plastique et obéir au principe de travail plastique maximal, ce qui veut dire, concernant les interfaces, que la loi de frottement doit respecter la règle de normalité (le critère de plasticité des interfaces étant convexe). Comme évoqué

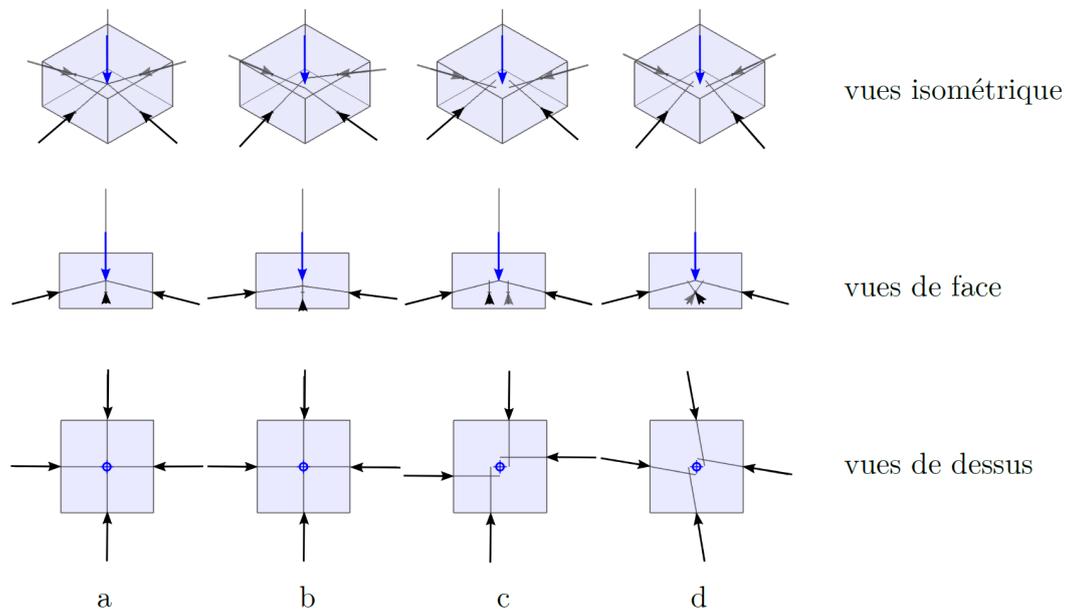


FIGURE 3.9 – Systèmes de forces en équilibre. Pour les cas (b), (c) et (d), les forces ne sont pas concourantes en un point. (Issue de Fantin (2017))

précédemment, la troisième hypothèse d’Heyman n’implique pas le respect de la règle de normalité. Bagi (2014) a fait ce constat et a proposé de séparer les déplacements au niveau des interfaces en deux groupes : heymaniens et non heymaniens (cf. Fig. 3.2). Bagi (2014) remplace ainsi la troisième hypothèse d’Heyman par :

3. Pas de déplacements non-Heymaniens entre les blocs

L’analyse limite de Heyman s’applique donc avec ces nouvelles hypothèses et protège seulement contre les déplacements Heymaniens.

Remarquons que les interfaces parfaitement rugueuses respectent la règle de normalité et la cinématique d’écoulement associée est celle des déplacements heymaniens (cf. Section 3.2.2). Une autre façon de voir cette troisième hypothèse est donc de modéliser le contact comme étant parfaitement rugueux.

Si la loi de frottement considérée ne respecte pas la règle de normalité, nous ne sommes plus dans le cadre de l’analyse limite mais dans celui du calcul à la rupture. Dans le cadre du calcul à la rupture, aucune donnée n’est nécessaire sur le matériau autre que sa résistance. L’inconvénient est que, là où l’analyse limite donne des chargements sûrement supportés, le calcul à la rupture donne des chargements seulement *potentiellement* supportables (Salençon, 1983, 2002). Il s’avère que dans la maçonnerie, les frottements sont plutôt de type Coulomb non associés (ne respectant pas la règle de normalité) (cf. Section 3.2.2) et que le calcul à la rupture surestime donc le chargement réellement supporté si la ruine peut avoir lieu selon des déplacements non-Heymaniens.

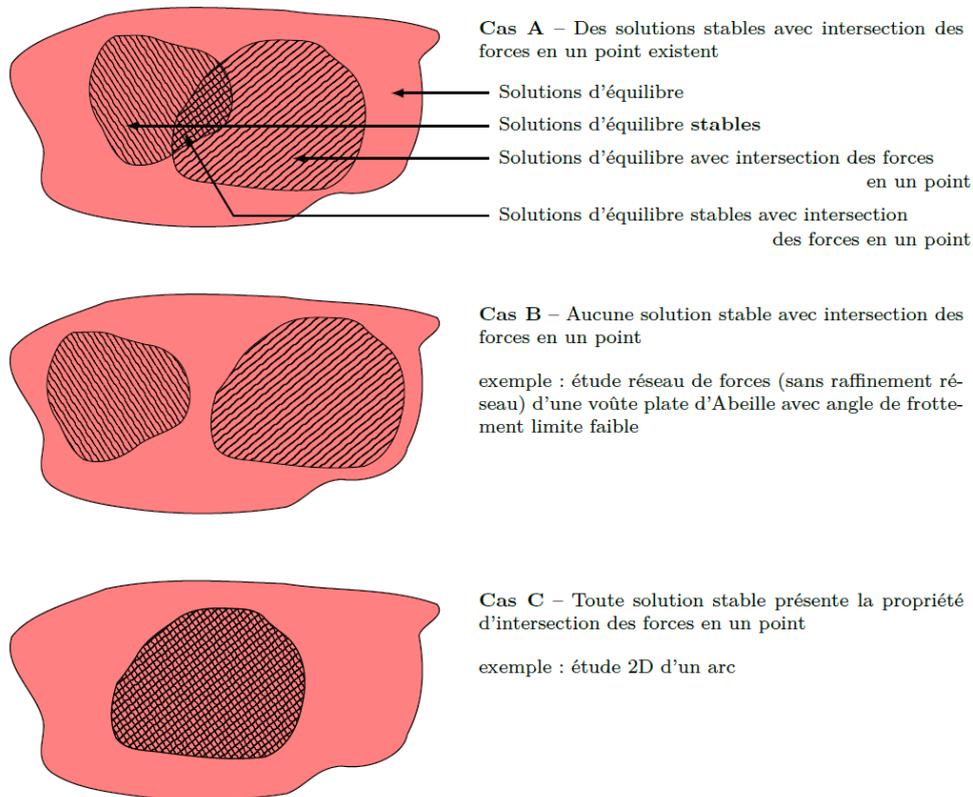


FIGURE 3.10 – Domaines des solutions d'équilibre ; intersections possibles des solutions d'équilibre stables, avec ou sans la propriété d'intersection des forces en un point (Fantin, 2017))

3.4.6 Conclusion

Pour calculer la stabilité des assemblages topologiquement autobloquants présentés dans la première partie, le calcul à la rupture ou l'analyse limite ne nous semble pas être un choix pertinent. D'une part, la recherche d'efforts statiquement admissibles est rendue difficile par la stéréotomie particulière de tels assemblages, particulièrement si la géométrie considérée est courbe. Par ailleurs, on souhaite s'intéresser aux interfaces et à leur inclinaison. Une ruine par glissement – ou plus généralement par déplacement non heymanien – n'étant pas exclue, le calcul à la rupture risque de surestimer le chargement limite supporté. Cette méthode a néanmoins l'avantage d'être paramétrable et de permettre la recherche de formes plus facilement que par des calculs aux éléments discrets.

3.5 Éléments finis continus

Une autre approche pour résoudre le problème hyperstatique dans l'analyse de stabilité des structures maçonnées est d'utiliser la théorie de l'élasticité notamment au travers du calcul par éléments finis en considérant la structure comme un milieu continu (sans modélisation géométrique des joints). Utiliser une analyse élastique linéaire ne rend pas bien compte de la grande sensibilité aux perturbations des structures maçonnées (Huerta, 2001; Block et al., 2006). La ligne de pression et le polygone funiculaire sont en effet très sensibles aux petits changements des conditions aux limites, ce qui ne transparaît pas forcément dans le calcul par éléments finis linéaires. Comme le montre la figure Fig. 3.11, l'arc en haut à gauche est instable car le polygone funiculaire est en dehors de l'épaisseur de la maçonnerie (c). L'arc en bas à gauche est quant à lui stable car il existe un polygone funiculaire compris dans l'épaisseur de l'arc (d). Pourtant le calcul par éléments finis linéaires – (a) et (b) – donne des résultats très similaires ne faisant pas ressortir aussi bien cette instabilité. De plus, des joints peuvent s'ouvrir, notamment lors du décintrage, et la structure rester néanmoins stable. C'est un autre type de phénomène non linéaire non considéré dans une analyse élastique linéaire. Le milieu continu de Cauchy est mal adapté pour représenter la spécificité de la maçonnerie, notamment la possible rotation des blocs n'est pas prise en compte. Une modélisation de la maçonnerie par milieu de Cosserat mène à de meilleurs résultats (Besdo, 1985; Masiani et al., 1995). Tout comme l'homogénéisation, faire une analyse avec des éléments finis de type Cosserat ne prend pas en compte de manière directe la géométrie des blocs et des interfaces. Une telle analyse n'est donc pas adaptée pour l'étude des assemblages topologiquement autobloquants lorsque la géométrie des blocs et l'inclinaison des joints est étudiée. Il est enfin possible d'effectuer un calcul par éléments finis en modélisant chaque joints par des éléments finis non linéaires de joints mais cette solution est mal adaptée aux cas de grands déplacements et de création de nouveaux contacts (Pegon et al., 2001; Giordano et al., 2002).

3.6 Méthode des éléments discrets

La méthode des éléments discrets (DEM) consiste à modéliser chaque bloc séparément, comme un corps distinct, (géométrie, lois constitutives, etc) ainsi que les lois de comportement régissant les contacts. Cette méthode a initialement été développée par Cundall (1971) sous le nom de "Distinct Element Method" pour l'étude 2D des masses rocheuses. Un programme peut être qualifié de programme d'éléments discrets s'il remplit les deux conditions suivantes (Cundall and Hart, 1992) :

- permet des grands déplacements et rotations de corps discrets ainsi que le détachement complet
- détecte automatiquement les nouveaux contacts au cours de l'analyse

3DEC (Cundall, 1988; Hart et al., 1988; 3DEC, 2013) est un programme par éléments discrets issu des travaux de Cundall (1971). Ce logiciel intègre en temps les équations

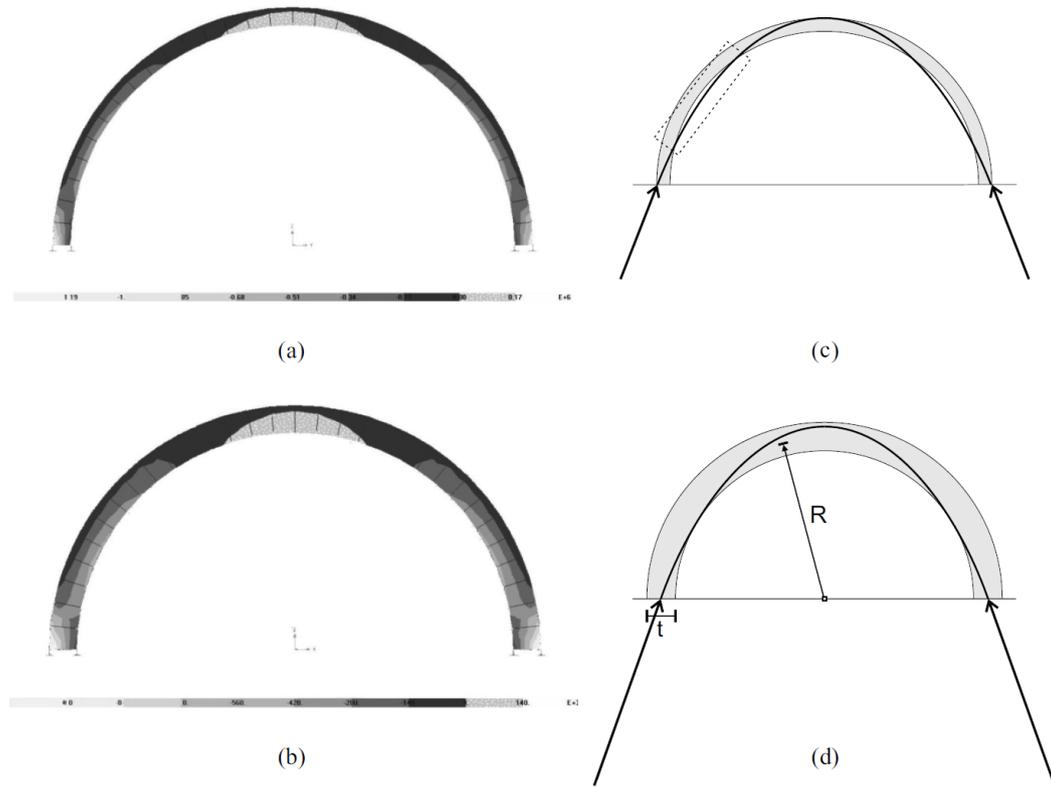


FIGURE 3.11 – Analyse d’un arc par éléments finis linéaires et par polygone funiculaire. À gauche sont représentés les champs de contraintes, et à droite, les polygones funiculaires. L’aspect des champs de contraintes représentés en (a) et (b) sont similaires et il n’est pas évident de conclure sur la stabilité des arcs à partir de ces données. En revanche, le polygone funiculaire sort de l’arc dans la figure (c) et pas dans la figure (d). L’arc (c) – correspondant à l’arc (a) – est donc instable. L’arc (d) – correspondant à l’arc (b) – est, a priori, stable. (Issue de Block et al. (2006))

de mouvements de chaque bloc de manière explicite grâce à la méthode des différences finies centrées. Lemos (2007) a fait un état de l’art des différents modèles se basant sur les éléments discrets pour l’étude des structures maçonnées. A retenir que les éléments peuvent être des corps polyédriques ou particuliers, les corps peuvent être considérés comme des solides rigides ou déformables (modélisé par un maillage d’éléments finis) et la formulation des contacts peut être du type *hard contact* (aucune interpénétration possible au niveau des joints) ou *soft contact* (élasticité au niveau des joints les rendant déformables). Puisque la méthode de résolution est basée sur une intégration en temps, pour analyser les équilibres statiques, des méthodes issues de la relaxation dynamique sont utilisées en ajoutant un amortissement artificiel grand.

Abaqus (Simulia, 2014), logiciel de calcul aux éléments finis, permet aussi désormais

de faire un calcul du type éléments discrets. Les études de stabilité présentées dans le Chapitre 5 ont été effectuées grâce à Abaqus en modélisant chaque bloc séparément et en utilisant une analyse explicite.

Citons enfin Brugger et al. (2008); Dugué (2013) qui ont développé un code d'éléments discrets pour les assemblages autobloquants où chaque bloc est un point matériel et les lois de contact inter-bloc sont modélisées grâce à un calcul par éléments finis. Cette méthode a été appliquée dans le cas où tous les blocs sont identiques. Si tous les blocs sont différents, il faudrait calculer les lois d'interaction inter-bloc par éléments finis pour chaque paire de blocs en contact.

3.7 A propos des moteurs physiques type PhysX

Pour calculer la stabilité d'une structure maçonnée lors de l'assemblage, Kao et al. (2017) utilise le moteur physique en temps réel PhysX de NVIDIA, traditionnellement utilisé dans l'industrie du jeu vidéo. Ce moteur physique se base sur la dynamique des corps rigides et peut simuler un frottement de type Coulomb. PhysX est comparé avec la méthode des éléments discrets (3DEC) de manière qualitative au travers de deux simulations, l'une de basculement d'un bloc en fonction de l'encorbellement d et l'autre de glissement d'un bloc en fonction de la pente θ (cf. Fig. 3.12). Les résultats sont similaires mais, proche de l'instabilité, 3DEC est meilleur.

Cependant quelques réserves peuvent être émises. La comparaison n'est que qualitative et concerne un système isostatique. Il serait intéressant de comparer PhysX et 3DEC sur une structure hyperstatique, comme par exemple dans Godio et al. (2018), pour avoir une idée de la validité des résultats, d'autant plus que l'industrie des jeux vidéos privilégie souvent la rapidité à la précision. Néanmoins, l'article de Kao et al. (2017) nous montre que c'est un outil utile pour des tests rapides. Grâce à la rapidité de la simulation via PhysX, on peut ainsi filtrer les configurations d'assemblages impossibles avant de vérifier avec 3DEC.

3.8 Conclusion

Plusieurs approches pour l'étude de stabilité des maçonneries ont été présentées dans ce chapitre. En particulier, l'analyse limite (ou plus généralement le calcul à la rupture), le calcul par éléments finis continus et celui par éléments discrets. L'analyse limite n'est pas appropriée pour l'analyse des maçonneries autobloquantes car la recherche de solutions d'équilibre est rendue difficile à cause de la stéréotomie particulière de ces maçonneries. De plus, elle surestime les chargements limites supportés car ne considère pas un critère de frottement de Coulomb non associé. Le calcul par éléments discrets est plus adapté pour cette analyse mais nécessite des données supplémentaires sur le comportement matériau, le comportement des joints, etc. Or ces données ne sont pas toujours connues ou facilement accessibles. L'analyse par éléments discrets se fait plutôt a posteriori pour vérifier la stabilité de la structure et n'est pas très bien adaptée (contrairement au TNA) pour une phase de conception. Néanmoins, nous utiliserons ce type d'analyse dans le

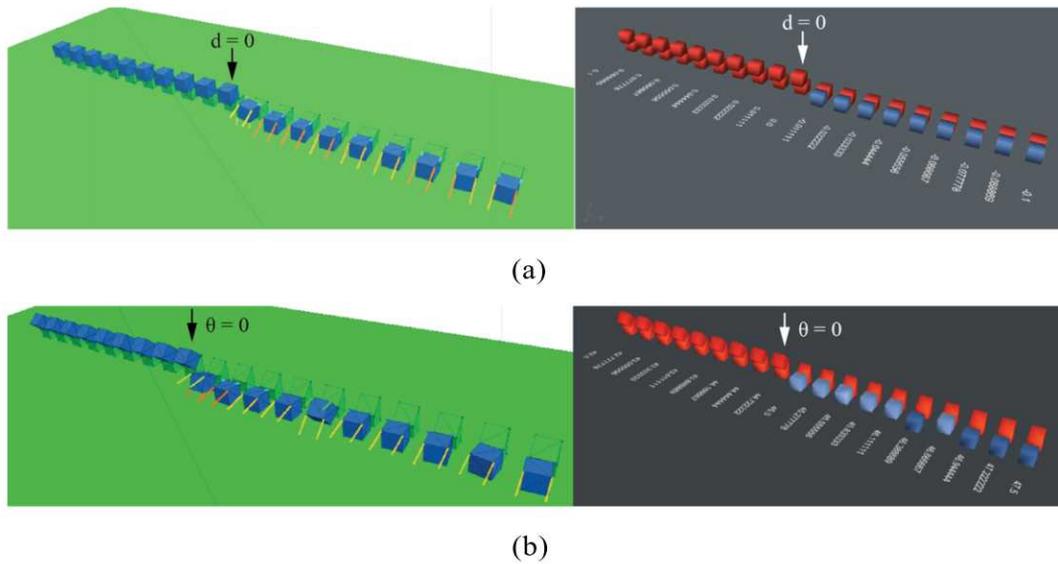


FIGURE 3.12 – Capture d’écran des deux expériences en utilisant 3DEC (gauche) et Rhinoceros 3D avec moteur PhysX (droite). Dans l’expérience (a), deux cubes sont superposés. Le cube du dessus est en porte-à-faux par rapport au cube du dessous. Le porte-à-faux est paramétré par la variable d . Plusieurs cas avec des valeurs de d différentes sont représentés. L’expérience (b) est similaire mais avec une pente d’inclinaison variable θ . (Issue de Kao et al. (2017))

Chapitre 5 pour vérifier la stabilité de maçonneries topologiquement autobloquantes courbes.

Chapitre 4

Vision artificielle et estimation de pose

Résumé : Ce chapitre s'intéresse aux méthodes possibles pour faire de la préhension robotisée de blocs sur un environnement de chantier grâce à la vision artificielle, ces blocs pouvant être les blocs topologiquement autobloquants présentés dans les chapitres précédents. L'apprentissage supervisé et les réseaux de neurones sont rapidement présentés, puis l'apprentissage à partir de données synthétiques est exposé. Enfin, on introduit diverses méthodes utilisées pour localiser et estimer la pose (position + orientation) d'objets dans des images 2D.

4.1 Apprentissage et réseaux de neurones

À l'attention des lecteurs qui ne sont pas familiers avec les notions d'apprentissage et de vision artificielle, nous passons en revues quelques concepts de base sans rentrer dans les détails techniques. Cette section n'a ainsi pas la vocation d'être exhaustive sur les réseaux de neurones artificiels mais vise seulement à introduire certains concepts utilisés dans cette thèse, tels que l'architecture d'un réseau, le surapprentissage ou les bases de données de validation.

4.1.1 Apprentissage automatique et modèle

L'apprentissage automatique (ou *machine learning*) est une technique d'intelligence artificielle. Il emploie plusieurs types d'algorithmes, dont notamment les **réseaux de neurones artificiels**.

L'apprentissage automatique consiste en l'estimation d'une fonction – communément appelée *modèle* – qui prédit une sortie à partir d'une entrée donnée. Si l'estimation est bonne, elle permet alors d'effectuer une certaine tâche qui lui est associée. Par exemple, dans le domaine de la vision artificielle, on peut souhaiter estimer un modèle qui prédit ce qui est présent dans une image. L'entrée du modèle est une image et la sortie est la classe de l'objet représenté dans l'image (un chien, un chat, un avion ou une chaise dans l'exemple de la Fig. 4.1). Si l'estimation du modèle est bonne, elle prédit avec succès ce qui se trouve dans l'image. Nous pouvons également citer, comme autre exemple, ce que nous faisons dans cette thèse : nous essayons d'estimer un modèle qui prédit le positionnement relatif d'un bloc par rapport à un bras robot, à partir d'une image de la scène où ceux-ci apparaissent.

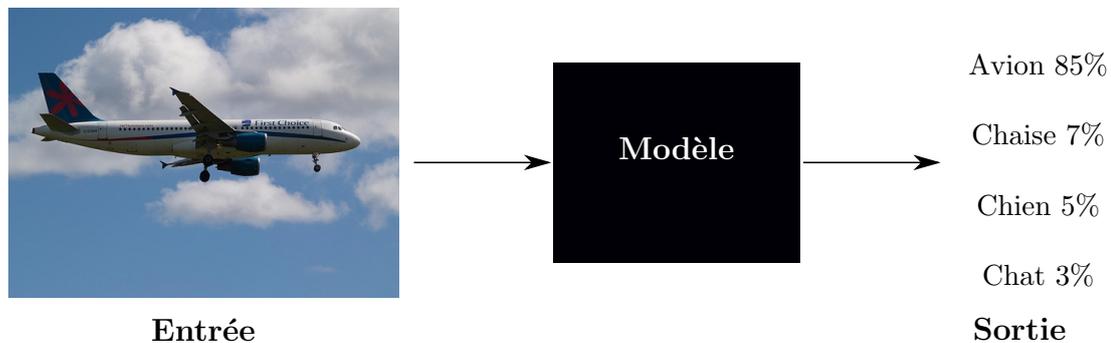


FIGURE 4.1 – Modèle effectuant de la classification d'image. Ici, le modèle prédit le degré de confiance, parmi les quatre classes disponibles, de ce qui se trouve dans l'image.

4.1.2 Apprentissage supervisé

4.1.2.1 Différents types d'apprentissages

Pour estimer un modèle, on peut se donner, a priori, une fonction f définie par un certains nombre de paramètres, appelés *poids* dans le cas des réseaux de neurones. Estimer le modèle revient alors à trouver des bonnes valeurs de paramètres pour que l'estimation effectuée bien la tâche demandée (à condition que la fonction f soit bien choisie). Pour cela, l'apprentissage automatique a d'abord recours à une première étape dite d'**apprentissage**. Il existe différents types d'apprentissage (supervisé, non supervisé, par renforcement, etc), mais nous nous intéressons ici uniquement à l'apprentissage supervisé. Celui-ci consiste à trouver les valeurs optimales des poids à partir d'une base de données d'apprentissage. Cette base de données est constituée de nombreuses données annotées (Fig. 4.2). L'algorithme d'apprentissage supervisé optimise alors la valeur des poids de la fonction en analysant ces données. L'apprentissage est dit supervisé, car pour une donnée d'entrée, le résultat souhaité est connu. Une fois le modèle bien entraîné, si on lui fournit en entrée une donnée qu'il n'a jamais vu, il doit être capable d'effectuer la tâche demandée ; par exemple, reconnaître effectivement ce qui se trouve dans une image qu'il n'a jamais vue auparavant.



FIGURE 4.2 – Exemple d'images issues de la base de données ImageNet. Chaque image comporte une annotation décrivant son contenu.

4.1.2.2 Exemple : régression linéaire

Trouver la droite de régression dans le cas d'une régression linéaire est un des exemples les plus simples d'apprentissage supervisé. Dans cet exemple, nous ne considérons plus des images mais le jeu de données annotées suivant, avec $x_i \in \mathbb{R}$ la donnée d'entrée et $y_i \in \mathbb{R}$ la donnée de sortie associée (Fig. 4.3).

$$\{(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \text{ avec } x_i \in \mathbb{R}, y_i \in \mathbb{R}$$

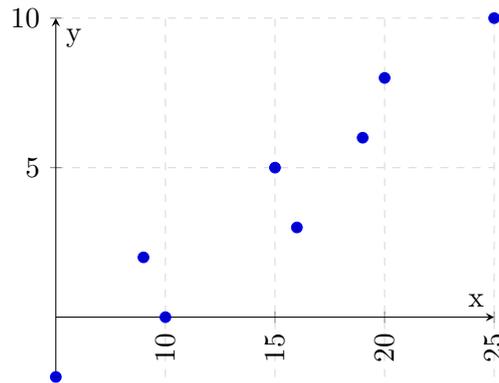


FIGURE 4.3 – Données d'apprentissage

Supposons que nous pouvons approximer le modèle par une fonction affine f :

$$f(x) = ax + b$$

avec a et b réels. a et b sont les paramètres (les poids) à déterminer à partir de la base de données. Si les données sont réparties approximativement le long d'une droite, la fonction f choisie, à deux paramètres, devrait être suffisante pour estimer le modèle.

Les paramètres sont bien estimés si $f(x_i)$ est "proche" de y_i pour tout i . Cette notion de proximité entre $f(x_i)$ et y_i se traduit mathématiquement par l'introduction d'une **fonction de coût**. Si les $f(x_i)$ et y_i sont proches, cette fonction coût aura une valeur faible. Inversement, elle aura une valeur élevée si $f(x_i)$ et y_i sont éloignés. Soit l cette fonction de coût, que nous considérons quadratique :

$$l(a, y) = \frac{1}{2}(a - y)^2$$

Le coût pour une des observations du jeu de données est donc

$$\begin{aligned} l(f(x_i), y_i) &= \frac{1}{2}(f(x_i) - y_i)^2 \\ &= \frac{1}{2}(ax_i + b - y_i)^2 \end{aligned}$$

Le coût total sur l'ensemble du jeu de données est la moyenne empirique :

$$\begin{aligned}
 L &= \frac{1}{n} \sum_{i=0}^n l(f(x_i), y_i) \\
 &= \frac{1}{n} \sum_{i=0}^n \frac{1}{2} (ax_i + b - y_i)^2
 \end{aligned}$$

Minimiser ce coût L permet alors de trouver les paramètres a^* et b^* optimaux. La fonction $f^*(x) = a^*x + b^*$ estime ainsi le modèle. Si l'estimation est bonne, alors pour un x quelconque de \mathbb{R} , $f^*(x)$ donne le résultat souhaité (Fig. 4.4).

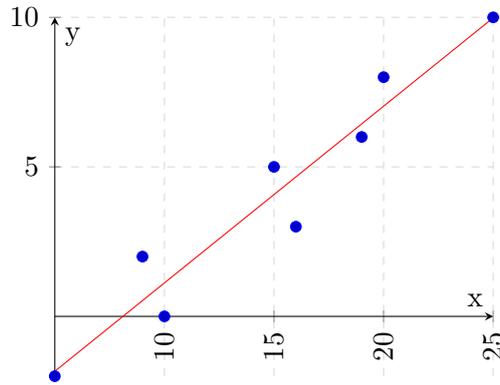


FIGURE 4.4 – La droite f^* approxime les données d'apprentissage

Remarque 1 : Ici, le problème pouvait être raisonnablement approximé par une fonction affine. Les modèles à estimer ne sont en revanche pas tous estimables par des fonctions linéaires et il peut être utile d'utiliser des fonctions non linéaires. Les réseaux de neurones, présentés ci-dessous, sont des fonctions non linéaires, contenant beaucoup de paramètres, permettant d'estimer des problèmes plus complexes que celui que l'on vient de présenter. Pour faire la classification d'images présentée Fig. 4.1, un algorithme d'apprentissage supervisé peut être également utilisé. Néanmoins, utiliser une fonction f affine ne donnera probablement pas de bons résultats. Les réseaux de neurones convolutifs sont de bons candidats de fonctions à optimiser pour faire de la classification d'images.

Remarque 2 : Les algorithmes d'apprentissage supervisé peuvent aussi bien servir à faire de la régression (prédire une valeur quantitative, numérique, d'une variable) que de la classification (prédire à quelle classe appartient la donnée d'entrée). Par exemple, l'exemple de la Fig. 4.1 est un exemple de classification. L'exemple que l'on vient de présenter, en revanche, est un exemple de régression. Néanmoins, les problèmes de régression peuvent souvent se formuler comme des problèmes de classification. Par exemple, supposons que l'on souhaite estimer un modèle qui prédit une distance y comprise entre 0 m et 1 m. Vu comme un problème de régression, le modèle prédit une unique valeur

y_{estimé}. On peut également considérer le problème comme un problème de classification en divisant le domaine $[0; 1[$ en plusieurs cases (*bins*) de taille fixée. Par exemple, avec une taille de case Δp , on obtient les cases $[0, \Delta p[$, $[\Delta p, 2\Delta p[$, ..., $[1 - \Delta p, 1[$. Le modèle doit alors prédire, par classification, la case correspondante à la donnée d'entrée.

4.1.3 Réseau de neurones artificiels

4.1.3.1 Neurone formel

Avant de présenter les réseaux de neurones utilisés en apprentissage, nous présentons leur inspiration biologique (Hubel and Wiesel, 1962).

Schématiquement, un neurone biologique (Fig. 4.5) reçoit des signaux électrochimiques au niveau de ses dendrites, ces signaux provenant des terminaisons axonales des neurones avoisinants. Si la somme de tous ces signaux atteint un certain seuil, le neurone entre dans un état excité permettant la propagation d'un influx nerveux le long de l'axone. Cet influx nerveux se transmet aux neurones suivants, via les terminaisons axonales, en excitant les dendrites de ces derniers.

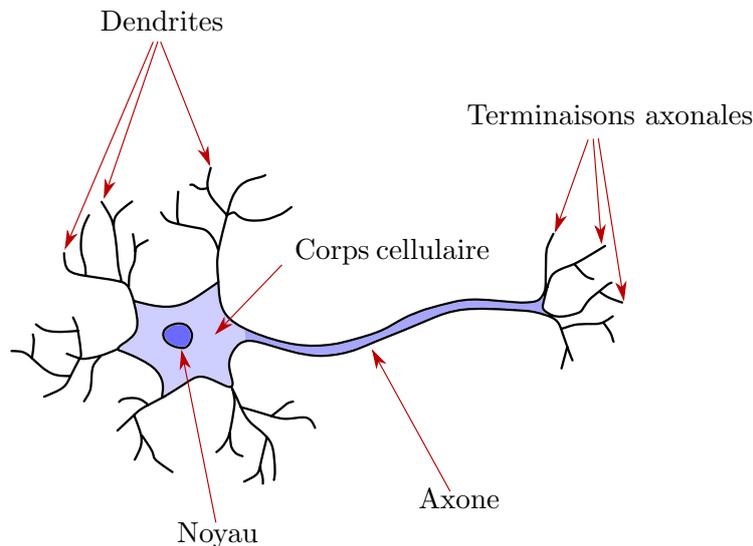


FIGURE 4.5 – Représentation d'un neurone biologique

Un neurone formel (Fig. 4.6) est une fonction mathématique qui s'inspire du neurone biologique. Les données d'entrée (x_0, x_1, \dots, x_n) sont pondérées par des poids (w_0, w_1, \dots, w_n) , puis sommées. Une fonction d'activation non linéaire – ici notée h – est ensuite appliquée à cette somme pondérée. La non-linéarité apportée par cette fonction procure au neurone formel un comportement similaire à celui d'un neurone biologique. En effet, si les signaux reçus par un neurone biologique l'excitent suffisamment, ce dernier propage un signal via son axone. C'est la même chose ici, si la somme pondérée dépasse un certain seuil, un signal non nul est propagé en sortie.

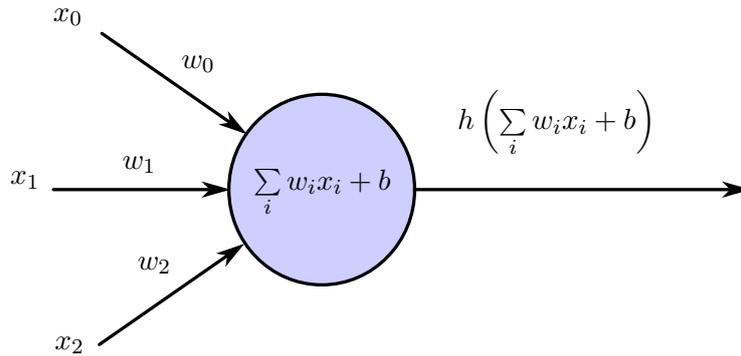


FIGURE 4.6 – Représentation d'un neurone formel.

Plusieurs fonctions d'activation ont été développées. Citons pour exemple la fonction sigmoïde (Fig. 4.7) ou la fonction Rectified Linear Unit (ReLU) (Fig. 4.8). C'est cette dernière qui est principalement utilisée, car, entre autre, elle accélère le temps de calcul.

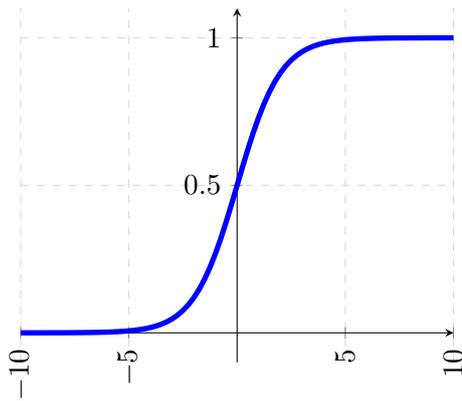


FIGURE 4.7 – Sigmoide

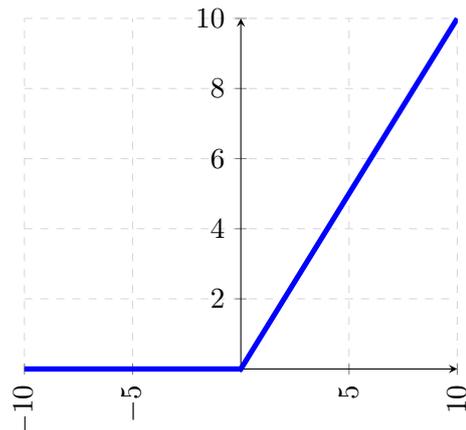


FIGURE 4.8 – ReLU

4.1.3.2 Réseaux de neurones

Considérons un seul neurone formel dont la fonction d'activation est une fonction de Heaviside – valant 0 en dessous d'un certain seuil et 1 au dessus. Ce neurone peut être vu comme un réseau de neurones à un seul neurone faisant de la classification binaire. Il s'appelle *perceptron* et a été inventé en 1957 par F. Rosenblatt (Rosenblatt, 1958). Le perceptron est une fonction qui contient plusieurs paramètres (les poids w_i), il peut jouer le rôle de la fonction f dans un algorithme d'apprentissage supervisé (Sec. 4.1.2).

Cependant, le perceptron est trop simple pour effectuer des tâches de classification complexes. D'où le principe de relier plusieurs neurones entre eux pour obtenir un réseau de neurones (voir Fig. 4.9), comme par exemple le Neocognitron (Fukushima, 1980) qui

inspira les réseaux de neurones convolutifs. Comme on peut le voir sur la Fig. 4.9, les neurones sont agencés par couche. Les neurones d'une même couche ne communiquent pas entre eux. Les sorties des neurones d'une couche sont les entrées des neurones de la couche suivante. Plus le nombre de couche est élevée, plus il y a de poids, et plus l'apprentissage – l'optimisation de la valeur des poids – est difficile. Dans ces réseaux, la sortie de certains neurones sont l'entrée d'autres neurones. Le terme d'apprentissage profond (*deep learning*) est régulièrement utilisé pour parler des réseaux contenant de nombreuses couches. L'**architecture** d'un réseau fait référence au nombre de neurones, au nombre de couches et à leur agencement dans le réseau. Il n'est pas nécessaire de concevoir une nouvelle architecture de réseau à chaque nouvelle tâche. Il est possible d'utiliser des réseaux qui donnent des bons résultats sur d'autres problèmes. Citons par exemple, AlexNet (Krizhevsky et al., 2012) qui a gagné en 2012 la compétition de classification d'images ILSVRC et est à l'origine de la révolution de l'apprentissage profond. Dans notre thèse, nous utilisons par exemple le réseau existant ResNet (He et al., 2015).

Réseaux de neurones convolutifs De manière simplifiée, les réseaux de neurones peuvent être considérés comme des fonctions contenant beaucoup de paramètres (poids). Ces fonctions peuvent être utilisées en apprentissage supervisé en mettant à jour la valeur de leurs poids en fonction d'exemples présents dans une base de données annotées. De telles fonctions se sont montrées particulièrement efficaces pour résoudre des problèmes de vision artificielle. Comme nous venons de le voir, un neurone formel correspond à une fonction linéaire (une somme pondérée) suivie de l'application d'une fonction non linéaire (la fonction d'activation). Pour adapter les réseaux de neurones à l'analyse d'images, la somme pondérée est remplacée par un produit de convolution (qui est encore une fonction linéaire). Nous ne rentrerons pas dans les détails ici, mais nous retiendrons seulement que les **réseaux de neurones convolutifs** (LeCun et al., 1989) sont des réseaux de neurones adaptés pour le traitement des images.

4.1.3.3 Descente de gradient stochastique et époques

La mise à jour de la valeur des poids du réseau se fait grâce à l'introduction d'une fonction de coût (parfois appelée fonction objectif) et à un algorithme d'optimisation. La fonction à optimiser n'est en général pas convexe et le nombre de variables (les paramètres) sur lesquelles effectuer l'optimisation est quant à lui très grand. Un algorithme du gradient n'est alors pas approprié. Pour optimiser les réseaux de neurones, un autre algorithme est utilisé, la **descente de gradient stochastique** (SGD). Cet algorithme est itératif, fait appel à la rétropropagation du gradient (Lecun et al., 1998) pour calculer le gradient de chaque neurone et permet de mettre à jour la valeur des poids du réseau en regardant, à chaque itération, seulement une partie des données d'apprentissage. C'est-à-dire qu'il n'est pas nécessaire, lors d'une itération, d'utiliser toute la base de données avant de mettre à jour la valeur des poids. À chaque itération, une partie seulement de la base de données est utilisée. À l'itération suivante, une autre partie de la base de données sera utilisée. Et ainsi de suite jusqu'à avoir utilisé tous les éléments

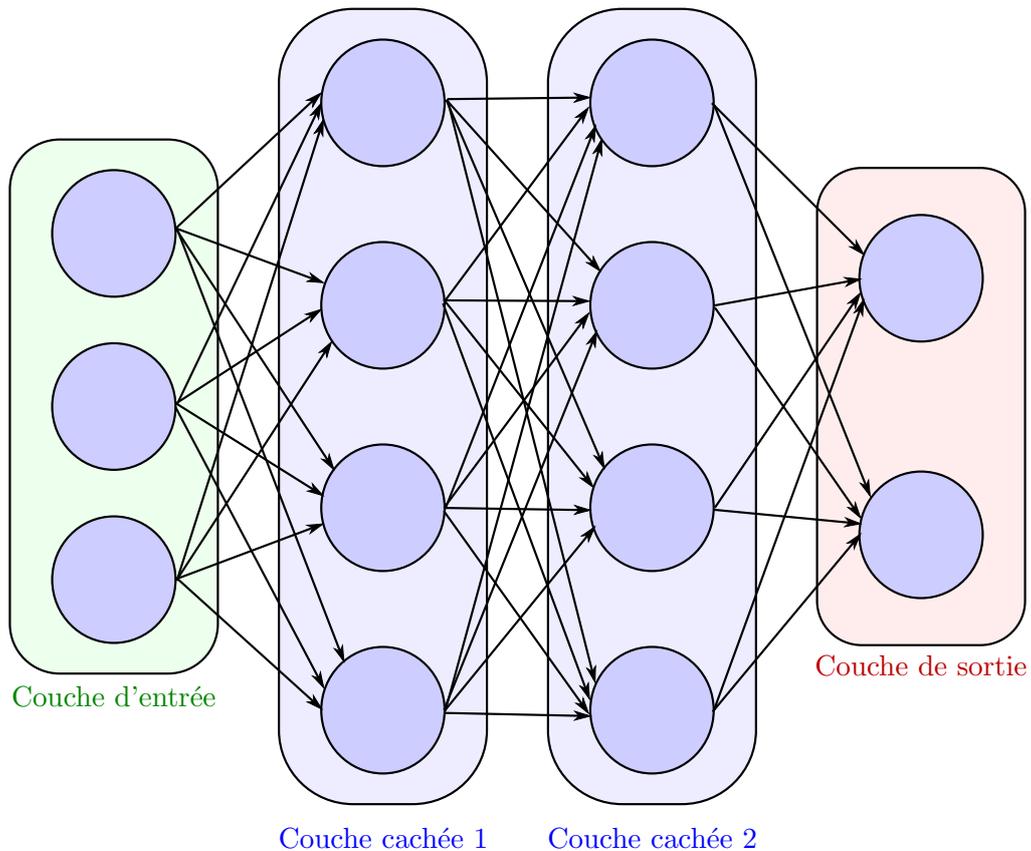


FIGURE 4.9 – Représentation d’un réseau de neurones à trois couches (la couche d’entrée n’est jamais comptée)

de la base de données. Une époque qualifie le fait d’avoir observé tous les éléments de la base de donnée une fois. À la fin d’une époque, toutes les données ont été utilisée une fois, mais l’algorithme d’optimisation n’a pas nécessairement convergé. Les données sont alors de nouveau utilisées lors d’une seconde époque et ainsi de suite. Pour optimiser la valeur des poids du réseau, il est généralement nécessaire d’utiliser toutes les données plusieurs fois, autrement dit, d’entraîner le réseau pendant plusieurs époques.

4.1.4 Surapprentissage

Un problème courant en apprentissage, et notamment dans le cas de réseaux de neurones, est celui du surapprentissage.

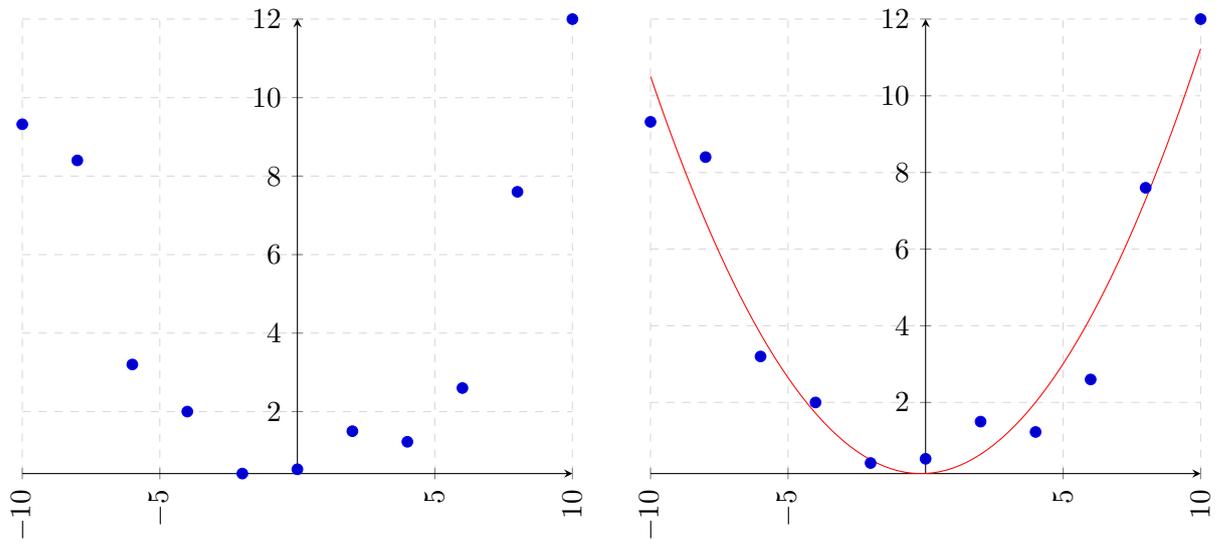
4.1.4.1 Exemple

Supposons que l'on effectue un apprentissage supervisé à partir du jeu de données représenté Fig. 4.10a. En effectuant l'apprentissage en utilisant un polynôme de degré 2, on obtient l'estimation de la Fig. 4.10b. On peut estimer que, qualitativement, ce polynôme estime bien le jeu de données. En effectuant l'apprentissage en utilisant un polynôme de degré 10, l'estimation de la Fig. 4.10c est obtenue. La courbe passe exactement par tous les points du jeu de données, mais qualitativement, nous pouvons considérer que l'estimation est plus mauvaise que celle effectuée par le polynôme de degré 2. Pour s'en convaincre, imaginons une nouvelle donnée d'abscisse -9, nous nous attendons à obtenir une ordonnée valant environ 9. L'interpolation d'ordre 2 donne un peu près ce résultat alors que celle d'ordre 10 donne un résultat proche de 13, soit pratiquement 50% d'erreur. Ce phénomène s'appelle le **surapprentissage**. Il survient lorsqu'il y a trop de paramètres à optimiser par rapport à la quantité de données d'apprentissage. Ici, la base de données contient 11 éléments. Le polynôme de degré 2 contient 3 paramètres à optimiser alors que celui de degré 10 en contient 11. Optimiser la valeurs de 11 paramètres à partir de 11 données a mené au surapprentissage.

4.1.4.2 Données d'entraînement et de validation

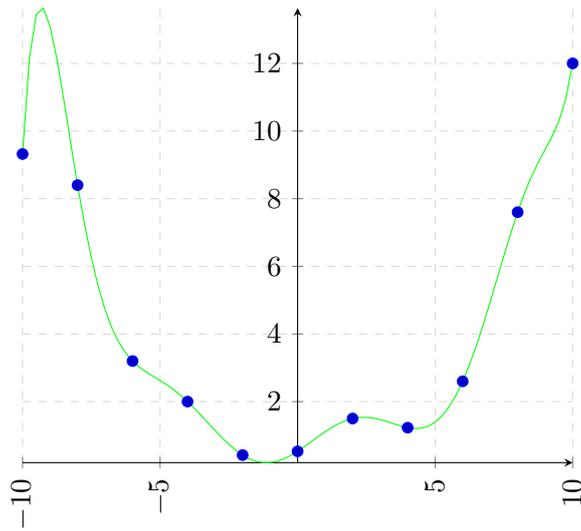
Afin de savoir s'il y a surapprentissage lors de l'entraînement, il est d'usage de diviser la base de données en deux parties. Une partie – *la base de données d'entraînement* – est utilisée pendant l'apprentissage pour mettre la valeur des poids à jour. L'autre partie – *la base de données de validation* – n'est jamais utilisée pour mettre la valeur des poids à jour, elle sert uniquement à vérifier que l'algorithme entraîné donne de bons résultats sur des données qu'il n'a jamais vues. Si l'erreur obtenue sur les données d'apprentissage et l'erreur obtenue sur celles de validation diminuent conjointement, l'algorithme apprend. Si l'erreur obtenue sur les données d'apprentissage diminue mais celle obtenue sur les données de validation reste grande et stagne, on peut soupçonner un cas de surapprentissage. Dans l'exemple de la Fig. 4.10c, l'erreur est nulle sur toutes les données d'apprentissage. L'erreur sur des données de validation serait en revanche non nulle. En revanche, dans l'exemple de la Fig. 4.10b, on peut s'attendre à avoir des erreurs d'entraînement et de validation du même ordre de grandeur (et plus petite que l'erreur de validation obtenue dans le cas du polynôme de degré 10).

Le surapprentissage surgit lorsqu'il y a trop de poids à optimiser et pas assez de données d'entraînement. Pour s'en prémunir, il faut diminuer le nombre de poids ou bien augmenter la taille de la base de données. Cependant, en diminuant le nombre de poids, on court le risque d'obtenir une fonction pas assez riche pour bien estimer le modèle ; par exemple, une régression linéaire (contenant 2 poids) donne des résultats médiocre sur l'exemple de la Fig. 4.10. Afin d'augmenter la taille de la base de données, pour éviter le phénomène de surapprentissage, nous pouvons avoir recours à l'augmentation de données (*data augmentation*). Par, exemple, l'image miroir d'une image de chien reste toujours une image de chien. Cependant, du point de vue de l'ordinateur, les deux



(a) Jeu de données

(b) Interpolation par un polynôme de degré 2



(c) Interpolation par un polynôme de degré 10

FIGURE 4.10 – Illustration du surapprentissage

images sont totalement différentes. D'autres méthodes de data augmentation sont aussi disponibles pour augmenter facilement la taille de la base de données, comme par exemple l'extraction aléatoire d'une sous-image de taille réduite (*random crop*). Elles sont très utiles lorsque l'on considère que le nombre de poids des réseaux de neurones utilisés se compte en plusieurs centaines de milliers, nécessitant alors une base de donnée de très grande taille.

4.1.5 À retenir

Les réseaux de neurones convolutifs sont généralement des fonctions contenant beaucoup de paramètres (poids). Les valeurs de ces paramètres sont trouvées lors d'une phase d'apprentissage supervisée afin que le réseau effectue bien la tâche demandée une fois entraîné. Pour effectuer cet apprentissage, il est nécessaire d'avoir une base de données annotées. Le nombre de données doit être d'autant plus grand que le nombre de paramètres (et le nombre de classes dans le cas d'un problème de classification) est grand. Si le nombre de données n'est pas assez grand, il peut y avoir surapprentissage. Pour vérifier qu'il n'y a pas eu de surapprentissage, le jeu de données est divisé en deux : une partie des données (données d'entraînement) sert vraiment à mettre à jour les poids. L'autre partie des données (données de validation) sert seulement à vérifier que le réseau prédit bien ce qui est voulu. Les bases de données que nous utilisons dans cette thèse sont constituées de plusieurs centaines de milliers d'images, que nous divisons en une base de données d'apprentissage et en une base de données de validation.

4.2 Apprentissage à partir de données synthétiques

Le succès de l'apprentissage profond a renouvelé l'intérêt de la possibilité d'apprendre à partir de données synthétiques. En effet, l'apprentissage profond requiert généralement de grandes bases de données annotées pour apprendre, mais de telles annotations sont souvent difficiles et coûteuses à produire. Or, des données synthétiques permettent d'avoir plus aisément des annotations précises pour chaque image. Cependant, un algorithme entraîné sur des données synthétiques n'est pas forcément efficace sur des données réelles, à cause de leurs nombreuses différences ; un problème auquel on fait souvent référence comme l'écart entre domaines différents (*domain gap*) entre images réelles et images synthétiques. Nous avons identifié trois approches principales qui tendent à résoudre ce problème.

4.2.1 Adaptation de domaine

L'apprentissage de modèle sur images synthétiques pour application à des images réelles peut être vu comme un exemple de problème d'adaptation de domaine¹, pour lequel de nombreuses approches ont été développées.

Une des approches est de construire une fonction de perte ou une architecture pour réaliser spécifiquement l'adaptation de domaine. De telles méthodes ont été utilisées sur des données synthétiques pour réaliser de la classification d'objets (Peng and Saenko,

1. De manière résumé, en apprentissage automatique, l'adaptation de domaine consiste à adapter l'apprentissage d'un modèle que l'on entraîne sur des données sources pour une utilisation sur des données cibles ; les domaines des données sources et cibles n'étant pas identiques. Ici nous entraînons un modèle à partir de données synthétiques mais nous l'utilisons sur des données réelles. A priori, puisque les deux domaines (d'apprentissage et d'utilisation) sont différents, rien ne prouve que le modèle appris sur l'un fonctionnera bien sur l'autre ; d'où l'utilisation de méthodes d'adaptation entre domaine pour "forcer" le modèle à donner de bons résultats sur les deux domaines.

2017), de la détection (Sun and Saenko, 2014; Vazquez et al., 2014) et, plus proche de notre problématique, pour de l’alignement de modèles 3D sur des images 2D (Massa et al., 2016b) et de l’estimation de pose (Rad et al., 2018). Il est également possible de forcer à apprendre des représentations indépendantes du domaine via une approche non supervisée (Ganin and Lempitsky, 2015). Enfin, l’écart entre domaines différents peut être réduit en entraînant deux réseaux différents, un pour chaque domaine (Girdhar et al., 2016; Grabner et al., 2018). Néanmoins, une limitation de cette approche est la difficulté d’avoir des bases de données annotées dans les deux domaines.

4.2.2 Simuler le réel

La façon la plus simple pour éviter les problèmes liés aux différences entre des images synthétiques et des images réelles est d’essayer de réduire l’écart entre domaines différents au maximum en générant des images les plus réalistes possible. Générer des images vraiment réalistes nécessite d’utiliser des modèles 3D de la scène de grande qualité ainsi qu’un bon modèle d’éclairage des objets physiques, puis d’appliquer un algorithme de rendu. Comme des modèles 3D de bonne qualité sont lourds et les rendus photoréalistes, à base de lancé de rayons, coûteux en temps de calcul (généralement plusieurs heures par image), peu de travaux ont été réalisés avec ce type de données et la plupart des travaux se concentrent plutôt sur l’imitation plus ou moins grossière du réel. Ceci peut être fait par exemple en fusionnant un rendu de modèle d’objet 3D avec une texture extraite d’une image réelle et en ajoutant un arrière-plan d’une scène réelle. En particulier, Peng et al. (2015) et Pepik et al. (2015) ont étudié comment les performances pour la détection d’objet étaient impactées par le réalisme de telles images. Su et al. (2015) ont appliqué cette stratégie pour améliorer les performances d’estimation de pose de catégories d’objets, et Chen et al. (2016) sur l’estimation de poses humaines. Plus récemment, de nombreux papiers ont utilisé des moteurs de rendu de jeux vidéos pour générer des données d’apprentissage, par exemple pour la segmentation sémantique (Richter et al., 2016; Ros et al., 2016; Shafaei et al., 2016).

4.2.3 *Domain randomization*

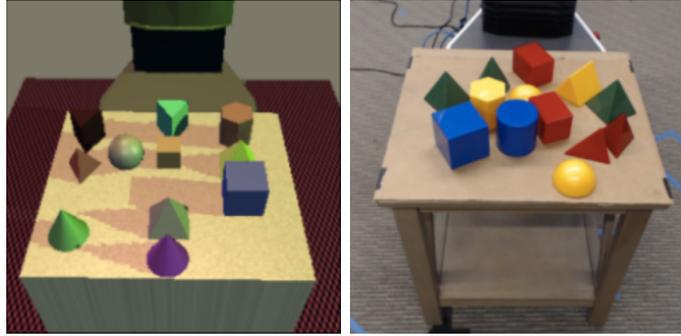
Une autre approche pour entraîner des modèles robustes avec des données synthétiques est de générer des exemples d’apprentissages d’une grande variété afin d’encourager les modèles à apprendre des invariances. En effet, les humains ont des connaissances préalables à propos du monde qui les entoure et les utilisent pour prendre des décisions. À l’inverse, une machine n’en a pas et il n’est alors pas nécessaire que la scène soit réaliste (Dubey et al., 2018).

La première application d’une telle approche avec des CNNs est probablement Dosovitskiy et al. (2015) qui tirent parti d’images composites – réalisées à partir d’images réelles et de rendus de chaises 3D (Aubry et al., 2014) – non réalistes et très modifiées (changements de luminosité, contraste, couleur, translation, rotation, etc) afin d’apprendre à prédire le flux optique entre des images réelles.

Tobin et al. (2017) ont récemment proposé une stratégie similaire, qu’ils ont appelé *do-*

main randomization, pour localiser un bloc et le saisir avec un robot. Il y a cependant plusieurs différences fortes entre ce travail et le nôtre. En particulier, Tobin et al. (2017) se concentrent sur la différenciation entre quelques formes d’objets (triangulaire, hexagonale, rectangulaire) et sur leur localisation grossière sur une table toujours observée d’un point de vue similaire avec la même orientation. Ainsi, cela revient essentiellement à localiser la table et les objets en 2D. A l’inverse, nous n’imposons pas la présence d’une table avec des caractéristiques fixes et un angle de vue donné; nous localisons directement les objets par rapport à un robot, vus par une ou plusieurs caméras, dans des positions inconnues. Ces différences de données sont illustrées dans la Fig. 4.11. En outre, notre approche permet de localiser le bloc avec une précision moyenne – pour une configuration avec une seule caméra – de 6.2 mm à 8.9 mm en position et 1.1° à 1.3° en orientation et une précision moyenne – pour une configuration à trois caméras – de 3.5 mm en position et 0.7° en orientation, alors que Tobin et al. (2017) obtiennent une précision moyenne de 15 mm, sans estimation de l’orientation.

Cette stratégie de *domain randomization* a été également utilisée récemment pour l’estimation de pose 6D par l’apprentissage d’auto-encodeurs sur des images synthétiques contenant beaucoup d’aléatoire (poses aléatoires, positions de l’éclairage et réflexions diffuse aléatoires, fond aléatoires, occlusions aléatoires, etc) (Sundermeyer et al., 2018); et, dans une moindre mesure, par Kehl et al. (2017), en incorporant des modèles 3D – dans des poses aléatoires et en variant le contraste et la luminosité – dans des images 2D pour entraîner un CNN de classification.



(a) Images synthétiques d'apprentissage et images réelles de validation issues de Tobin et al. (2017).



(b) Nos images d'apprentissage synthétiques et nos images réelles de validation.

FIGURE 4.11 – Comparaison qualitative des travaux récents de Tobin et al. (2017) avec notre approche. Alors que Tobin et al. (2017) se concentre sur la différence entre des objets sur une table de dimension fixe, nous localisons un bloc par rapport à un robot dans une configuration inconnue et dans un environnement inconnu, observé par une caméra dans une pose inconnue.

4.3 Localisation d’un objet et estimation de pose

Localiser un objet et estimer sa pose sont des problèmes classiques de vision par ordinateur. En effet, la compréhension 3D du monde visible dans les images semble être une première étape naturelle pour la compréhension de celles-ci et pour n’importe quelle interaction robotique (Roberts, 1963). La localisation consiste à trouver l’objet d’intérêt dans une image et à déterminer sa position. En général, cette localisation est représentée sous la forme d’une *bounding box* autour de l’objet. L’estimation de pose, quant à elle, consiste à estimer la position et l’orientation de l’objet par rapport à un système de coordonnées donné (souvent celui lié à la caméra). La localisation et l’estimation de pose peuvent être effectués séparément ou bien conjointement. Nous nous restreignons ici à la localisation et l’estimation de pose à partir d’une unique image RGB. Nous ne considérons donc pas l’estimation à partir de la profondeur – les caméras RGB-D ne fonctionnant pas bien en extérieur ou sur des objets métalliques.

4.3.1 Base de données pour alignement 2D-3D

Afin de pouvoir comparer les méthodes d’alignement, diverses bases de données ont été construites dans lesquelles un modèle 3D est aligné sur une image 2D. Le modèle 3D peut correspondre exactement à l’objet représenté dans l’image ou bien y correspondre seulement partiellement. Ainsi, l’alignement ne peut être qu’approximatif dans des bases de données comme Pascal3D+ (Xiang et al., 2014) ou ObjectNet3D (Xiang et al., 2016) puisque le modèle 3D n’est pas exactement celui de l’objet représenté dans l’image. Par exemple, ces bases de données proposent un alignement d’un modèle 3D de chaise sur une image contenant une chaise de même style, mais pas exactement similaire. En revanche, dans IKEA (Lim et al., 2013), le modèle 3D correspond effectivement à l’objet représenté dans l’image, l’alignement est précis mais la base de données ne contient que 759 images. Récemment, Pix3D (Sun et al., 2018) propose une base de données composée de plus de 10000 paires “images - objet 3D”, l’objet 3D correspond exactement à celui dans l’image et la précision de l’alignement est de l’ordre du pixel. Ces bases de données permettent ainsi de comparer les méthodes d’alignement 2D-3D mais ne s’appliquent néanmoins pas à notre cas car nous faisons de l’estimation de pose relative. Nous avons donc construit une nouvelle base de données contenant environ 12000 images réelles du bras robot et d’un bloc parallélépipédique, précisément localisés, afin d’évaluer quantitativement notre approche.

4.3.2 Méthodes basées sur l’alignement de modèle

Dans les premiers travaux de vision par ordinateur, il était souvent supposé que le modèle 3D de l’objet d’intérêt était disponible (Roberts, 1963; Lowe, 1987; Huttenlocher and Ullman, 1990; Mundy, 2006), le but étant de trouver les bonnes valeurs de position et d’orientation, par rapport à la caméra, pour l’aligner avec sa représentation dans l’image 2D. L’idée est de trouver des caractéristique locales de l’objet 3D à mettre en correspondance avec des caractéristiques locales de l’image 2D.

Les premières approches consistaient par exemple à extraire des caractéristiques géométriques de l'image (arêtes, coins géométriques, etc), puis à les regrouper pour obtenir des frontières d'objets. Il était alors possible de trouver la pose du modèle 3D qui correspond le mieux à ces regroupements de caractéristiques comme illustré Fig. 4.12. Néanmoins, l'identification des caractéristiques géométriques n'est pas aisée et ces méthodes se basant sur la géométrie de l'objet laissèrent alors la place à des méthodes plus efficaces utilisant des caractéristiques plus visuelles.

L'approche dominante était celle utilisant des descripteurs locaux de points-clés tels que les points SIFT (Lowe, 1999) illustré Fig. 4.13. Ces descripteurs sont localement invariants à l'échelle et à la rotation de l'image, et relativement robustes aux changements d'illumination ou de point de vue 3D. Cette approche a mené en particulier à de nombreuses chaînes de traitement d'estimation de pose 3D pour des applications robotiques, à partir d'une unique ou de plusieurs images (Collet and Srinivasa, 2010; Collet et al., 2011). Ces méthodes fonctionnent bien si l'objet est texturé car cela permet d'avoir beaucoup de points-clés. Inversement, si l'objet est peu texturé, la méthode sera moins efficace.

Une autre approche est de placer un ou plusieurs marqueurs visuels, tels que ArUco (Garrido-Jurado et al., 2014, 2016) sur les objets de la scène afin de faciliter leur détection et leur localisation précise dans les images. Cette méthode a été utilisée avec succès dans Feng et al. (2014) pour la construction robotisée d'un mur maçonné courbe (cf. Fig. 4.14). Elle est cependant moins flexible car elle nécessite de positionner avec soin et à des endroits bien définis les marqueurs sur des surfaces planes et visibles de l'objet. De plus, elle nécessite aussi la calibration extrinsèque et intrinsèque des caméras pour localiser le robot dans un référentiel global (potentiellement en plaçant aussi des marqueurs sur le robot) et pour établir les rapports entre les différents objets en 3D.

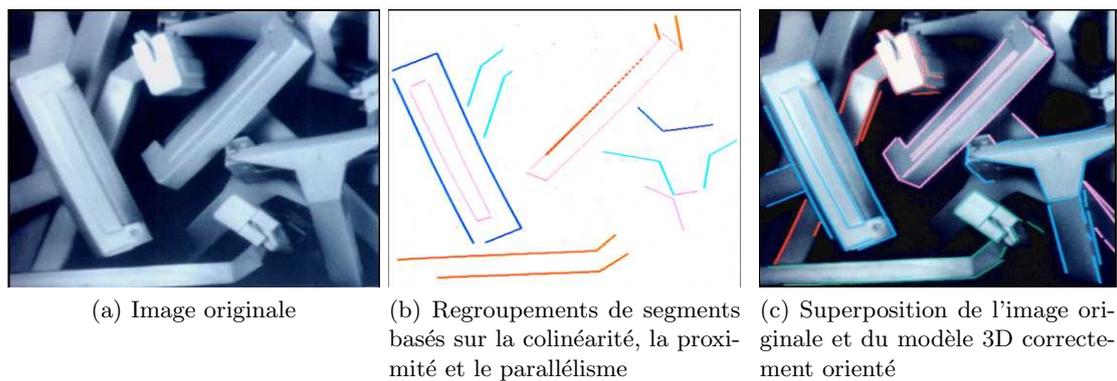


FIGURE 4.12 – Exemple de méthode basée sur l’alignement de modèle 3D sur image 2D d’après Lowe (1987).

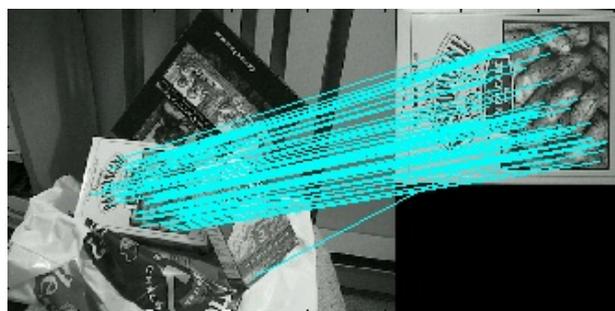


FIGURE 4.13 – Exemple de correspondances SIFT entre le modèle (à droite) et l’image (à gauche). (Image issue de <https://www.cs.ubc.ca/~lowe/keypoints/> [consulté le 3 décembre 2018])

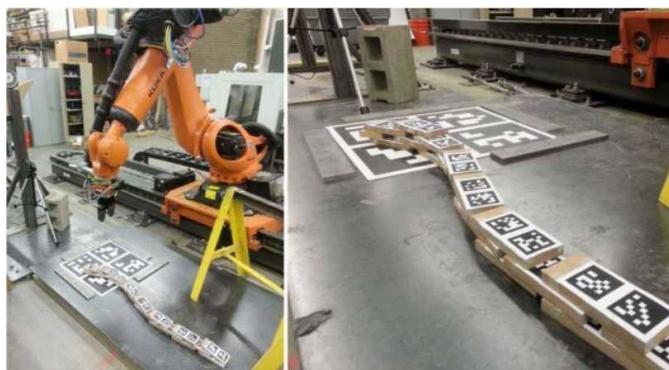


FIGURE 4.14 – Assemblage robotisé d’un mur courbe à l’aide de marqueurs visuels et de caméras calibrées (Feng et al., 2014).

4.3.3 Méthodes basées sur l'apprentissage

Les approches basées sur l'apprentissage tendent à se concentrer plus sur des catégories d'objets que sur un exemplaire d'un objet spécifique (*instance*). Elles peuvent approximativement être classées en deux catégories, celles ayant recours à des méthodes d'apprentissage "classiques" et celles recourant à l'apprentissage profond.

D'abord, de nombreux travaux utilisent des représentations (*features*) conçues à la main en tant qu'entrée de l'algorithme d'apprentissage et les appliquent à la détection d'objets dans des images. Par exemple, pour détecter des humains dans des images 2D, Dalal and Triggs (2005) entraînent un algorithme d'apprentissage à partir d'une représentation de l'image par histogramme de gradients orientés (HOG) et non directement à partir de l'image. Citons également les *Deformable Part Models* (DPM) (Felzenszwalb et al., 2010) qui fonctionnent particulièrement bien pour la détection (cf. Fig. 4.15).

L'usage des DPM a également été étendu à la prédiction de poses 3D d'objet (Glasner et al., 2011; Fidler et al., 2012; Hejrati and Ramanan, 2012; Pepik et al., 2012). Plus récemment, des méthodes d'apprentissage supervisé ont donné de bons résultats pour la localisation et l'estimation de pose à partir d'une image monoculaire (Choy et al., 2015). Mottaghi et al. (2015) ont développé une approche de détection avec une représentation en trois niveaux (grossière à fine), grâce aux SVM, permettant de prédire conjointement la localisation et l'estimation de pose 3D des catégories présentes dans PASCAL3D+ (Xiang et al., 2014). Ces approches sont appliquées à des catégories d'objets. Lim et al. (2013), en revanche, localisent et estiment la pose d'objets dans des images grâce à leur modèle 3D exact, grâce à une analyse discriminant linéaire (LDA).

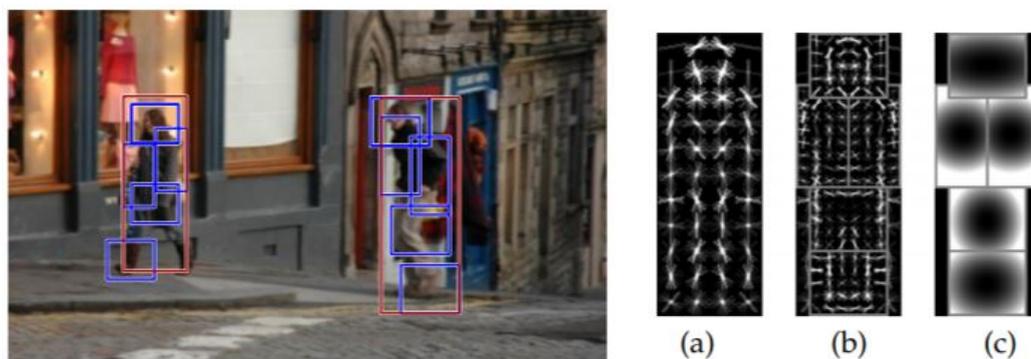


FIGURE 4.15 – Localisation d'objet avec les *Deformable Part Models*. (Image issue de (Felzenszwalb et al., 2010))

Il a été montré que les réseaux de neurones convolutifs (CNN) (LeCun et al., 1989) peuvent améliorer drastiquement les performances pour la localisation d'objets (Sermanet et al., 2014; He et al., 2017), notamment grâce aux R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015) et Faster RCNN (Ren et al., 2015) qui font usage de pro-

positions de régions (*region proposals*) pour détecter rapidement la localisation d’objets d’intérêts, comme illustré Fig. 4.16.

Outre la localisation, les CNNs permettent également d’estimer la pose d’objets (Tulsiani and Malik, 2015; Su et al., 2015; Wu et al., 2016; Massa et al., 2016a), en utilisant une approche par régression (Xiang et al., 2016) ou par classification (Tulsiani and Malik, 2015). Novotny et al. (2017) utilisent également les CNNs mais de manière non supervisée à partir d’un film, pour pouvoir ensuite estimer le point de vue et la profondeur à partir d’une seule image. Des méthodes plus récentes localisent et estiment la pose d’exemplaire d’un objet spécifique en alignant des modèles 3D rigides sur des images RGB. Ils utilisent la projection 2D de certains points de contrôle, afin d’estimer la pose de divers points d’intérêt de l’objet (Crivellaro et al., 2015) ou de sa *bounding box* (Rad and Lepetit, 2017). L’utilisation de divers points d’intérêts locaux rend par ailleurs la méthode robuste aux occultations.

Concernant des problématiques plus proches des nôtres, de telles approches d’apprentissage profond ont été utilisées par Pinto and Gupta (2016) pour apprendre à saisir un objet et ont aussi intéressé la communauté robotique, en particulier via l’apprentissage par renforcement (Schulman et al., 2015; Levine et al., 2016).

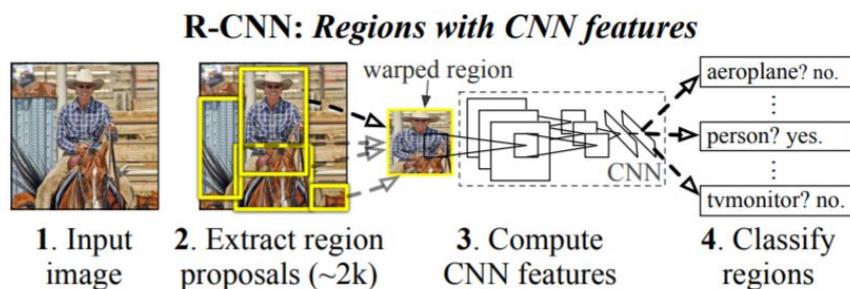


FIGURE 4.16 – Méthode de localisation d’objet R-CNN (Image issue de Girshick et al. (2014))

4.3.4 Position relative

Contrairement à ces travaux, notre but n’est pas uniquement la localisation d’un objet d’intérêt dans une image ou par rapport à la caméra, mais le positionnement relatif d’un objet par rapport à un autre; dans cette thèse, il s’agit de positionner un bloc par rapport à un bras robot (Chapitre 6). Plutôt que de viser une pose à 6 degrés de liberté (Hodañ et al., 2016; Kehl et al., 2017; Sundermeyer et al., 2018), nous nous restreignons à trois paramètres, deux degrés de liberté de position et un paramètre d’angle, puisque l’on peut raisonnablement supposer que le robot et le bloc sont tous les deux sur une surface relativement plane. Nous avons développé une approche de prédiction directe, en utilisant les progrès des approches basées sur les CNNs et suivant la logique de l’apprentissage de bout en bout, pour apprendre à prédire directement les

poses relatives. La définition du problème d'estimation de pose relative, ainsi que cette approche directe, sont des nouveautés de nos travaux.

4.4 Conclusion

Dans ce chapitre, nous avons présenté brièvement l'apprentissage supervisé et notamment les réseaux de neurones. Puisque nous apprenons à partir de données synthétiques, nous avons exposé différentes approches pour adapter l'apprentissage entre domaines différents. Enfin, nous avons présenté diverses techniques de localisation et d'estimation de pose.

Deuxième partie
Contributions

Chapitre 5

Assemblage topologiquement autobloquant sur surface courbe et condition de blocage

Résumé : Dans ce chapitre, nous exposons une procédure pour générer des assemblages topologiquement autobloquants à partir de pavages réguliers du plan. Nous généralisons ensuite cette procédure à des pavages non réguliers de surfaces courbes. Dans cette généralisation, les assemblages obtenus ne sont plus nécessairement topologiquement autobloquants. Nous étudions plus précisément les propriétés de blocage en translation et introduisons certaines relations pour que les assemblages obtenus par la généralisation de la procédure soient effectivement autobloquants. La notion d'assemblage fortement topologiquement autobloquant est également introduite et nous exposons un cas d'assemblage où chaque bloc est bloqué en translation, mais un sous-ensemble – constitué de plusieurs de ces blocs – ne l'est pas. Enfin, cette généralisation sera appliquée à des cas concrets et nous exposerons deux prototypes de maçonnerie obtenus grâce à celle-ci, un de ces prototype étant constructible sans utilisation de cintre.

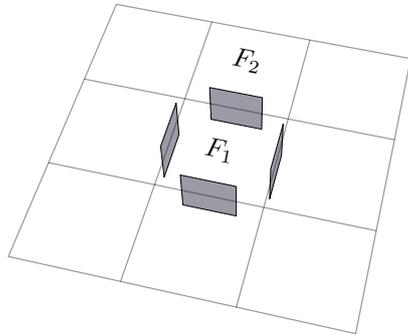
5.1 Méthode de construction d’assemblages topologiquement autobloquants à partir d’un maillage de quadrangles plans

5.1.1 *Moving cross-section procedure* et voûte plate d’Abeille

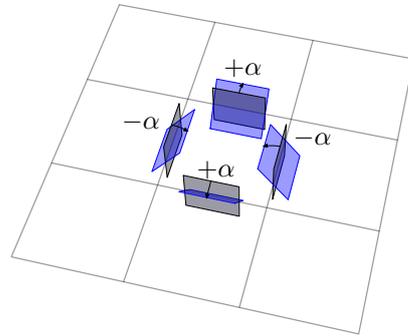
Nous décrivons ici la procédure *moving cross-section* mentionnée en Sec. 2.2.2, et illustrons comment elle permet d’obtenir des assemblages topologiquement autobloquants, en particulier, la voûte plate d’Abeille. Kanel-Belov et al. (2010) emploie cette procédure pour trouver des assemblages topologiquement autobloquants constitués des solides de Platon en utilisant, entre autres, des pavages réguliers du plan par des carrés ou par des hexagones – un pavage régulier du plan par des triangles ne permettant pas d’obtenir d’assemblages topologiquement autobloquants (Dyskin et al., 2003b).

La Fig. 5.1 illustre cette procédure avec un pavage/maillage constitué de carrés. Considérons une face de ce maillage et quatre plans qui lui sont perpendiculaires, chaque plan contenant une arête de la face (Fig. 5.1a). Ces plans sont inclinés alternativement avec des angles de valeurs $+\alpha$ et $-\alpha$ (Fig. 5.1b). Ces quatre plans inclinés renferment un volume ; ce volume étant un tétraèdre (Fig. 5.1c) que l’on peut éventuellement tronquer avec des plans parallèles au pavage initial (Fig. 5.1d) afin d’obtenir un tétraèdre tronqué (Fig. 5.1e). En répétant ces opérations pour chaque face du maillage, on obtient une voûte plate d’Abeille (Fig. 5.1f). Notons qu’une arête quelconque de ce maillage peut appartenir à deux faces voisines, par exemple F_1 et F_2 . Ainsi, pour des raisons de compatibilité, si le plan appartenant à la face F_1 et contenant cette arête est incliné d’un angle d’une valeur $+\alpha$, alors le plan appartenant à la face F_2 et contenant cette même arête est nécessairement incliné d’une valeur $-\alpha$. Autrement dit, dans cette procédure de construction d’assemblage topologiquement autobloquant, il y a un seul plan incliné par arête du maillage et ce plan incliné sert à construire à la fois le bloc issu de la face F_1 et le bloc issu de la face F_2 .

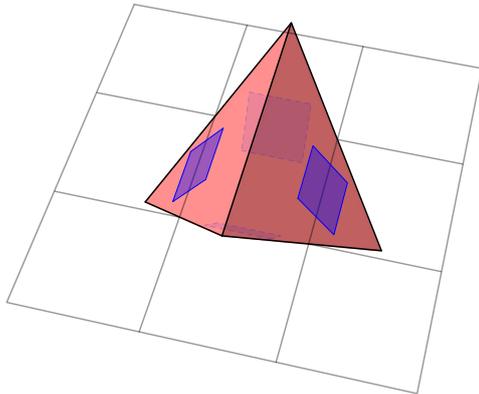
Soit un plan parallèle au pavage (par exemple, un des plans représentés Fig. 5.1d). L’intersection du solide (ici du tétraèdre) et de ce plan donne un polygone (ici un rectangle). En déplaçant ce plan de coupe le long de la normale au pavage initiale, le polygone se dégénère en une ligne. Ici, le carré du plan initial devient un rectangle puis se dégénère en une ligne. Dans le cas de l’assemblage de cubes de la Fig. 2.8, le polygone est un hexagone qui se dégénère en un point. Le nom de la procédure *moving cross-section* vient de là. Kanel-Belov et al. (2010) montre que si les polygones issus de la coupe avec des plans, de part et d’autre du pavage initial, se dégénèrent en une ligne ou un point, alors le bloc est bloqué.



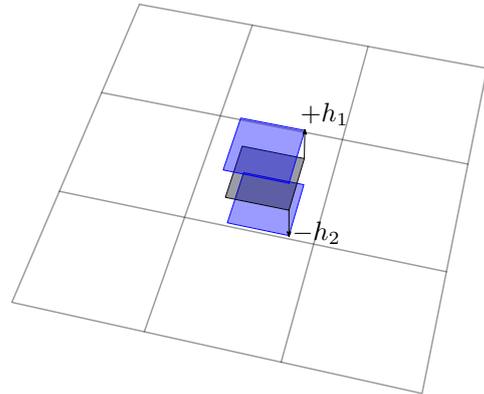
(a)



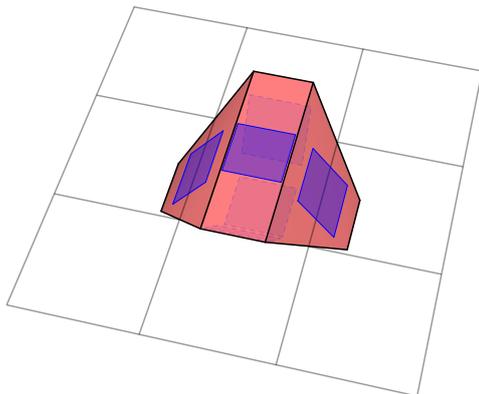
(b)



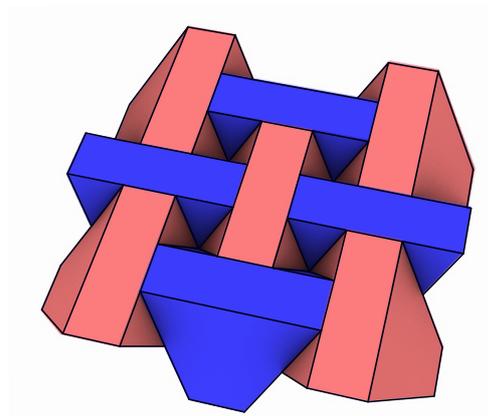
(c)



(d)



(e)



(f)

FIGURE 5.1 – Construction de la voûte plate d'Abeille

La procédure *moving cross-section* que l'on vient de décrire permet d'obtenir des assemblages topologiquement autobloquants à partir de pavages réguliers du plan :

1. associant des plans (perpendiculaires au plan du maillage) à chaque arête du maillage.
2. inclinant ces plans autour de leur arête respective avec, alternativement, des angles de valeur $+\alpha$ et $-\alpha$. Notons que la norme de ces angles est toujours égale à $|\alpha|$ avec $0 \leq |\alpha| < 90^\circ$.
3. tronquant éventuellement le solide obtenu par des plans parallèles au plan du maillage initial.

Nous souhaitons généraliser cette procédure à des pavages non réguliers de surfaces non nécessairement planes et en inclinant les plans avec des angles de normes non nécessairement égales. Pour cela, introduisons certaines notations, en considérant le cas particulier de pavage par des quadrangles plans¹. Nous verrons par la suite que cette généralisation de la procédure ne conduit pas forcément à des assemblages topologiquement autobloquants et que certaines inégalités doivent être respectées pour qu'ils le soient effectivement. Vella and Kotnik (2016) généralise également cette procédure pour des maillages de quadrangles quelconques sans néanmoins exprimer les conditions de blocage que nous exposerons.

5.1.2 Notations et conventions d'orientation

Soit \mathfrak{M} un maillage de quadrangles plans à partir duquel nous souhaitons construire un assemblage topologiquement autobloquant. Un exemple d'un tel maillage est représenté Fig. 5.2.

5.1.2.1 Notations locales

Soit F un quadrangle plan (une face) de ce maillage (Fig. 5.3). Notons \vec{n}_F sa normale unitaire et numérotions ses arêtes par les indices i , avec $i \in \{1, 2, 3, 4\}$. Soient \vec{t}_i le vecteur directeur unitaire de l'arête i et \vec{n}_{i0} le vecteur unitaire de telle sorte que $(\vec{n}_F, \vec{t}_i, \vec{n}_{i0})$ forment un trièdre direct. Par construction, les vecteurs \vec{n}_{10} , \vec{n}_{20} , \vec{n}_{30} et \vec{n}_{40} , s'appuyant sur le milieu de leur arête respective, sont inclus dans le plan du quadrangle et dirigés vers l'intérieur de celui-ci (Fig. 5.3a).

Soit P_{i0} le plan de normale \vec{n}_{i0} s'appuyant sur l'arête i . On obtient le plan P_i de normale \vec{n}_i par rotation d'angle α_i du plan P_{i0} autour de l'axe \vec{t}_i avec $0^\circ \leq |\alpha_i| < 90^\circ$ (Fig. 5.3b et Fig. 5.3c). Soit P_{h_1} (resp. P_{h_2}) le plan obtenu par translation de vecteur $h_1\vec{n}_F$ (resp. $h_2\vec{n}_F$) du plan du quadrangle F et \vec{n}_{h_1} (resp. \vec{n}_{h_2}) sa normale unitaire

1. Un quadrangle est un polygone à quatre angles, il a quatre sommets qui ne sont pas nécessairement coplanaires (trois points arbitraires de l'espace sont toujours coplanaires, ce n'est pas le cas pour quatre). Si les quatre sommets sont coplanaires, alors le quadrangle est plan. Un carré est un cas particulier de quadrangle plan.

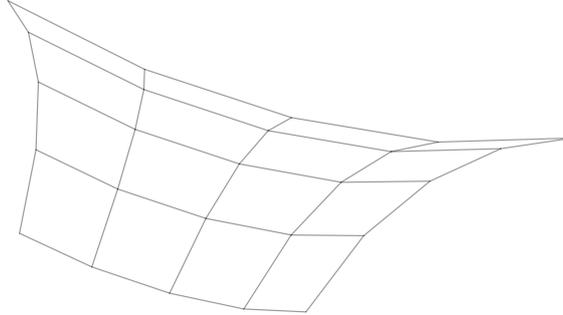


FIGURE 5.2 – Exemple de maillage d’une surface courbe par des quadrangles plans.

(Fig. 5.3d).

Chaque plan P_i divise l’espace euclidien canonique E de dimension 3 en deux demi-espaces. Un demi-espace E_i^+ du côté positif du plan orienté par la normale \vec{n}_i et un demi-espace E_i^- du côté négatif (Fig. 5.4). L’intersection des demi-espaces positifs associés aux plans $\{P_1, P_2, P_3, P_4\}$ donne un volume V , non nécessairement fini :

$$V = \bigcap_{i \in \{1,2,3,4\}} E_i^+$$

Dans l’exemple de la voûte d’Abeille de la Sec. 5.1.1, le volume V est fini et correspond au tétraèdre (Fig. 5.1c).

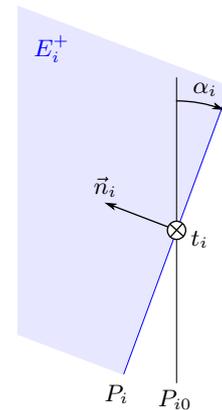


FIGURE 5.4

Si le volume V est non fini, on le rend fini en le limitant par les plans P_{h_1} et P_{h_2} . S’il est déjà fini, on peut néanmoins aussi le tronquer par ces plans.

Les plans P_{h_1} et P_{h_2} n’ont pas le même rôle que les plans $\{P_1, P_2, P_3, P_4\}$. Comme nous le verrons en Sec. 5.2.1, seuls ces derniers participent au blocage du bloc alors que P_{h_1} et P_{h_2} ne sont pas toujours nécessaires, mais jouent néanmoins un rôle important (Sec. 5.1.4). Ainsi, par la suite et par simplification, nous ne mentionnerons pas nécessairement les plans de troncatures P_{h_1} et P_{h_2} et nous parlerons donc de V comme du bloc associé à la face F et aux plans $\{P_1, P_2, P_3, P_4\}$.

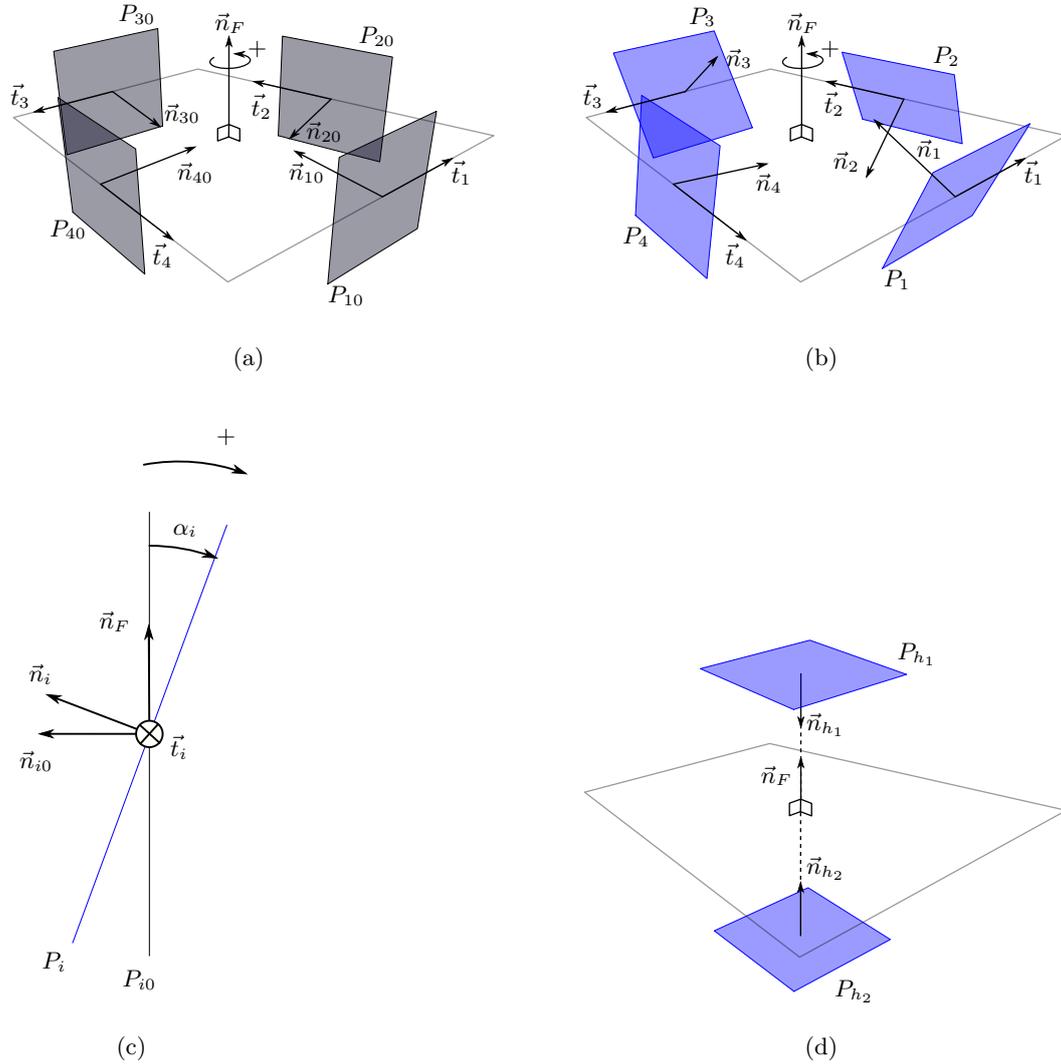


FIGURE 5.3 – Notations et conventions d'orientation

5.1.2.2 Notations globales et compatibilité

Les notations que l'on vient d'introduire sont des notations locales, elles ne concernent à chaque fois qu'une face du maillage. Or une arête e du maillage \mathfrak{M} peut appartenir à deux faces différentes F_1 et F_2 . Pour la face F_1 , l'indice local de l'arête est i et, pour la face F_2 , l'indice local est j .

Le bloc associé à la face F_1 et celui associé à la face F_2 doivent être compatibles entre eux, c'est-à-dire qu'ils doivent être en contact plan. Nous rappelons que cela revient à dire qu'il y a un unique plan incliné par arête e et donc $P_i^{F_1} = P_j^{F_2}$ avec $\vec{n}_i^{F_1} = -\vec{n}_j^{F_2}$. Nous nommons P^e cet unique plan incliné associé à l'arête e avec :

$$P^e = P_i^{F_1} = P_j^{F_2}$$

5.1.3 Généralisation de la procédure à des maillages de surfaces courbes

Dans le cas des pavages réguliers du plan, la première étape de la procédure consiste à associer des plans – que nous notons désormais P^{e0} – à chaque arête e du maillage, ces plans étant perpendiculaires au plan du maillage (Fig. 5.1a). Or nous considérons désormais des maillages de surfaces courbes. Il est encore possible d'associer un plan à chaque arête du maillage mais il faut redéfinir ce que l'on entend par *perpendiculaires au plan du maillage*.

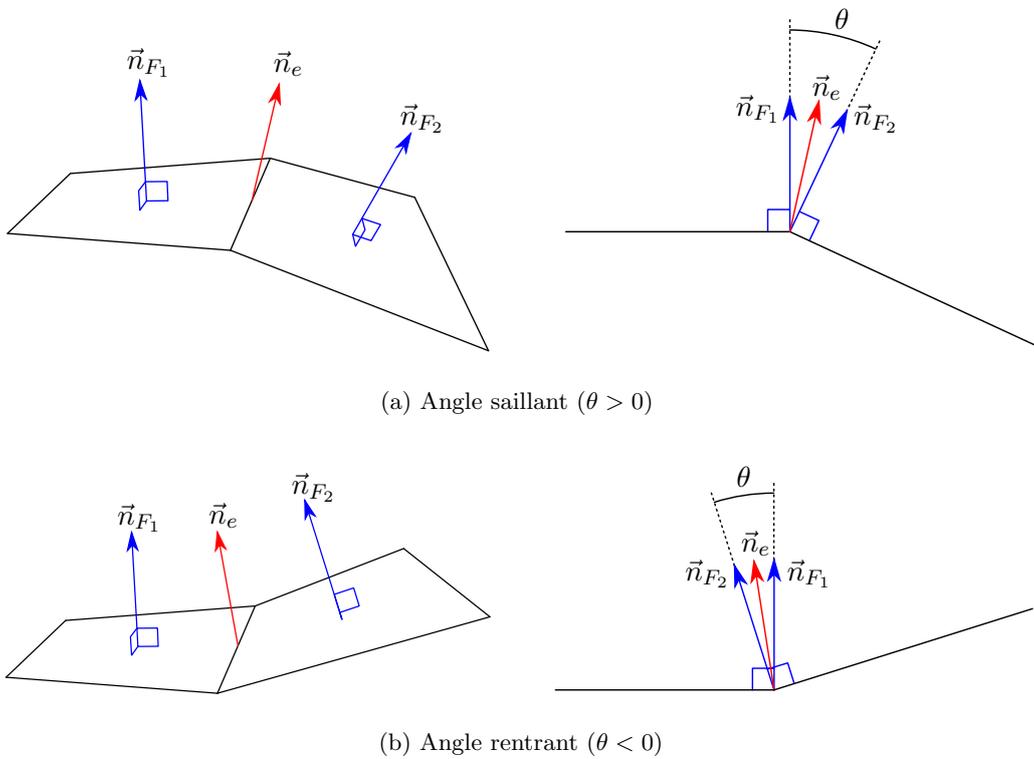


FIGURE 5.5 – Vecteur normal à une arête et angle diédral

Soient deux faces (du maillage) voisines F_1 et F_2 , de normale respective \vec{n}_{F_1} et \vec{n}_{F_2} (Fig. 5.5), partageant une arête commune e . Soit θ l'angle non orienté entre les deux vecteurs normaux \vec{n}_{F_1} et \vec{n}_{F_2} , nommé angle diédral. Pour $\theta > 0$, l'angle est saillant (Fig. 5.5a) et pour $\theta < 0$, l'angle est rentrant (Fig. 5.5b). Cet angle nous permet de définir la normale à l'arête \vec{n}_e :

$$\vec{n}_e = \frac{\vec{n}_{F_1} + \vec{n}_{F_2}}{\|\vec{n}_{F_1} + \vec{n}_{F_2}\|} = \frac{\vec{n}_{F_1} + \vec{n}_{F_2}}{2 \cos \frac{\theta}{2}}$$

Nous pouvons alors généraliser la première étape de la procédure par :

1. associer un plan P^{e0} à chaque arête e du maillage, ce plan contenant l'arête e et la normale à l'arête \vec{n}_e (Fig. 5.6).

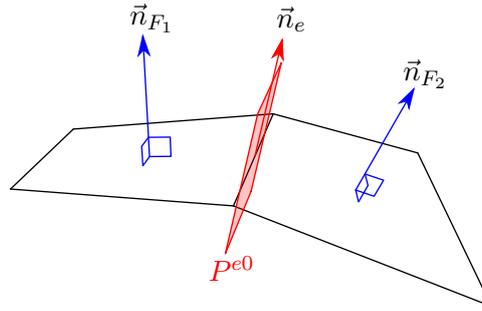


FIGURE 5.6 – Le plan P^{e0} contient l'arête e et le vecteur \vec{n}_e .

Lorsque les deux faces sont coplanaires ($\theta = 0$), on a $P^{e0} = P_{i0}^{F_1} = P_{j0}^{F_2}$. À l'inverse, lorsque l'angle diédral est non nul ($\theta \neq 0$), on a $P_{i0}^{F_1} \neq P_{j0}^{F_2}$ (Fig. 5.7a). Le plan P^{e0} se déduit en fait des plans $P_{i0}^{F_1}$ et $P_{j0}^{F_2}$ par une rotation de valeur $\frac{\theta}{2}$ autour de l'axe de l'arête e (Fig. 5.7b).

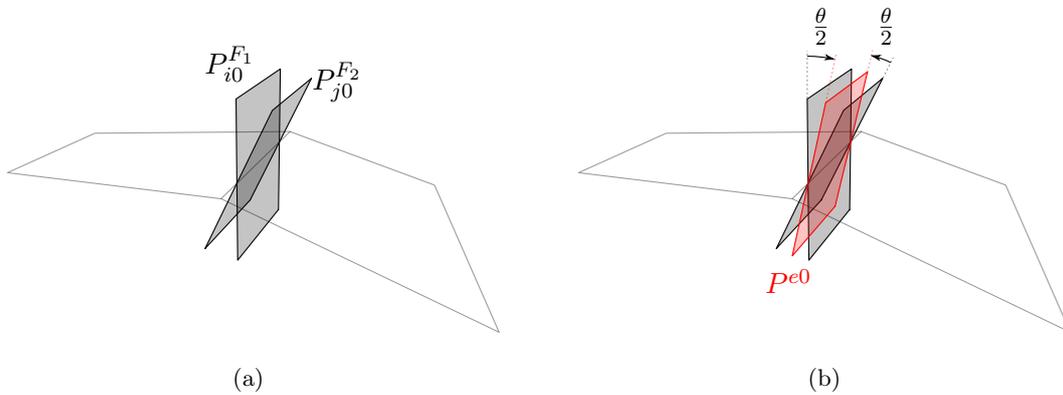


FIGURE 5.7 – Pour un angle diédral non nul (ici $\theta > 0$), les plans $P_{i0}^{F_1}$ et $P_{j0}^{F_2}$ ne sont plus confondus.

Dans la procédure pour des pavages réguliers du plan, la deuxième étape consiste à incliner chaque plan d'une valeur $\pm\alpha$, avec $0 \leq |\alpha| < 90^\circ$. Dans la généralisation de la procédure, nous souhaitons incliner chaque plan P^{e0} d'une valeur $\pm\beta^e$ pour obtenir les

plans P^e . La valeur de β^e pouvant être différente pour chaque plan.

Nous rappelons les trois propositions suivantes :

- $P^e = P_i^{F_1}$
- $P_i^{F_1}$ s'obtient par la rotation du plan $P_{i0}^{F_1}$ de valeur $\alpha_i^{F_1}$.
- P^{e0} s'obtient par la rotation du plan $P_{i0}^{F_1}$ de valeur $\frac{\theta}{2}$.

Nous avons alors, en considérant les notations locales relatives à la face F_1 (Fig. 5.8a) :

$$P^e \text{ s'obtient par la rotation du plan } P^{e0} \text{ de valeur } \beta_i^{F_1} = \alpha_i^{F_1} - \frac{\theta}{2}$$

De manière similaire, en considérant les notations locales relatives à la face F_2 (Fig. 5.8b) :

$$P^e \text{ s'obtient par la rotation du plan } P^{e0} \text{ de valeur } \beta_j^{F_2} = \alpha_j^{F_2} - \frac{\theta}{2}$$

De plus, puisque $P^e = P_i^{F_1} = P_j^{F_2}$, alors :

$$\beta_i^{F_1} = -\beta_j^{F_2}$$

Notons ainsi :

$$\beta^e = |\beta_i^{F_1}| = |\beta_j^{F_2}|$$

Enfin, puisque $|\alpha_i^{F_1}| < 90^\circ$ et $|\alpha_j^{F_2}| < 90^\circ$, nous avons la restriction suivante :

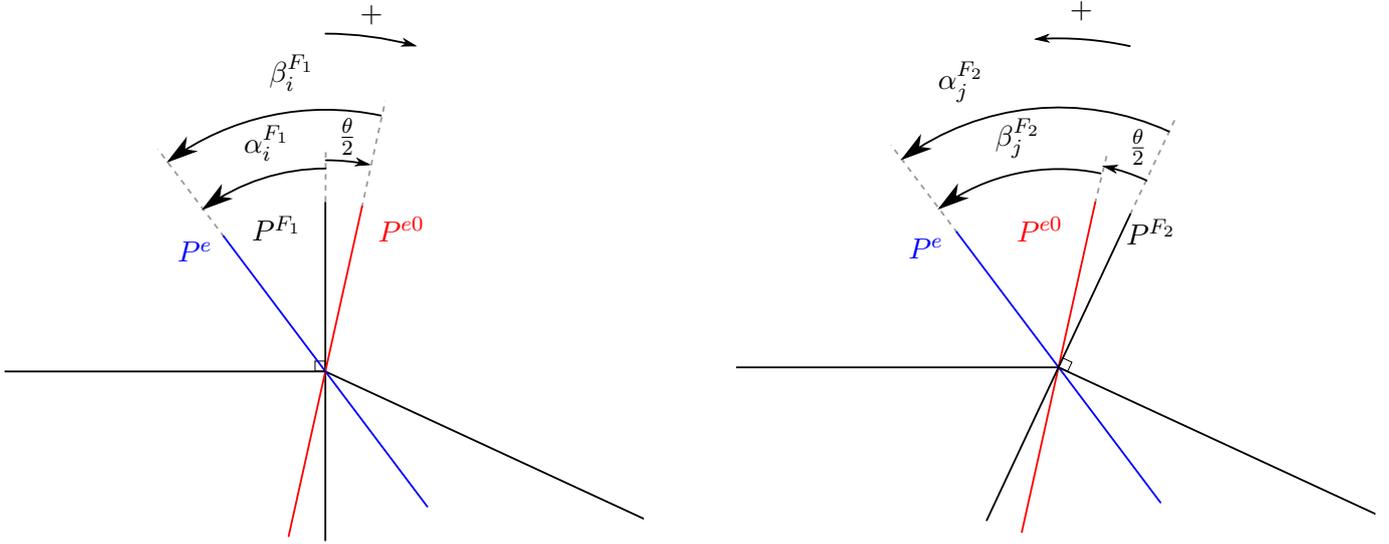
$$0 \leq \beta^e < 90^\circ - \frac{\theta}{2}$$

En effet, si cette inégalité n'est pas respectée, on a $|\alpha_i^{F_1}| > 90^\circ$ ou $|\alpha_j^{F_2}| > 90^\circ$, ce qui est interdit.

Explications Supposons $\theta > 0$. Puisque $\beta_i^{F_1} = -\beta_j^{F_2}$, si l'un des deux est positif, l'autre est négatif; et inversement. Nous pouvons alors supposer que $\beta_i^{F_1} > 0$. Ainsi, $\beta^e = |\beta_i^{F_1}| = \beta_i^{F_1} = \alpha_i^{F_1} - \frac{\theta}{2}$. L'inégalité précédente devient alors :

$$\alpha_i^{F_1} - \frac{\theta}{2} < 90^\circ - \frac{\theta}{2}$$

Si elle n'est pas respectée, $\alpha_i^{F_1} \geq 90^\circ$, ce qui n'est pas possible. On trouve des résultats similaires avec $\theta < 0$.



(a) Avec les notations locales et la convention d'orientation relatives à F_1 , on a ici : $\beta_i^{F_1} < 0$, $\alpha_i^{F_1} < 0$ et $\theta > 0$.

(b) Avec les notations locales et la convention d'orientation relatives à F_2 , on a ici : $\beta_j^{F_2} > 0$, $\alpha_j^{F_2} > 0$ et $\theta > 0$.

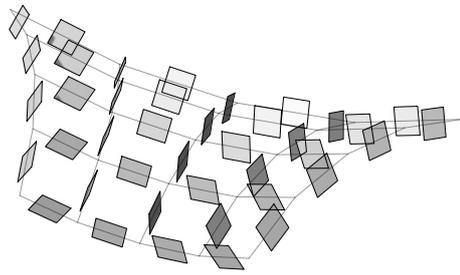
FIGURE 5.8 – Relation entre α , β et θ . L'égalité $\beta_i^{F_1} = -\beta_j^{F_2}$ est bien vérifiée.

La deuxième étape de la généralisation de la procédure est alors :

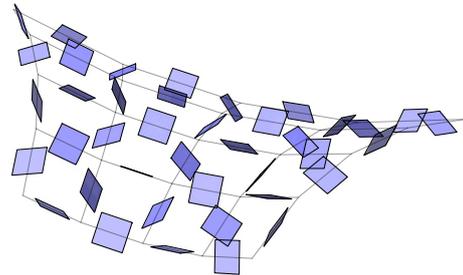
2. incliner les plans P^{e0} autour de leur arête respective avec des angles $\pm\beta^e$ vérifiant $0 \leq \beta^e < 90^\circ - \left|\frac{\theta}{2}\right|$.

La généralisation de la procédure décrite en Sec. 5.1.1 à des pavages non réguliers de surfaces non planes consiste donc à :

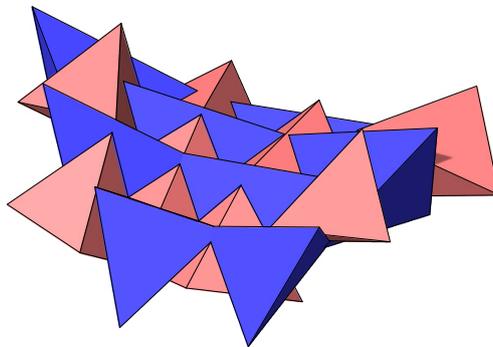
1. associer un plan P^{e0} à chaque arête e du maillage \mathfrak{M} , ce plan contenant l'arête e et la normale à l'arête \vec{n}_e (Fig. 5.9a).
2. incliner les plans P^{e0} autour de leur arête respective avec des angles $\pm\beta^e$ vérifiant $0 \leq \beta^e < 90^\circ - \left|\frac{\theta}{2}\right|$. (Fig. 5.9b).
3. pour chaque face F du maillage, obtenir le bloc V associé aux plans $\{P_1^F, P_2^F, P_3^F, P_4^F\}$ (Fig. 5.9c).
4. éventuellement tronquer chaque bloc V .



(a) Construction des plans P^{e0}



(b) Les plans P^e représentés ici sont obtenus en inclinant les plans P^{e0}



(c) Assemblage obtenu avant troncature

FIGURE 5.9 – Construction pavage de tétraèdres sur surface courbe. Ici, les angles β^e sont identiques pour chaque arête. Ce n'est pas obligatoire.

Remarque 1 Soit une face F du maillage formée par les arêtes e_1, e_2, e_3 et e_4 . Les plans $P^{e_1}, P^{e_2}, P^{e_3}$ et P^{e_4} sont obtenus respectivement par la rotation d'angle $\beta_1^F, \beta_2^F, \beta_3^F$ et β_4^F des plans $P^{e_1^0}, P^{e_2^0}, P^{e_3^0}$ et $P^{e_4^0}$. Pour chaque face de l'assemblage de la Fig. 5.9, ces angles sont de signes alternés et de normes égales : $+\beta_1^F = -\beta_2^F = +\beta_3^F = -\beta_4^F$. Ce n'est pas obligatoire. Nous verrons dans la suite de ce chapitre plusieurs exemples avec des angles de normes non égales et de signes non alternés.

Remarque 2 Il peut arriver, si les quadrangles sont dégénérés ou pour certaines valeurs d'inclinaison des plans P_e , que les blocs ainsi construits ne soient pas compatibles entre eux. Un bloc donné et un de ses voisins diagonaux peuvent en effet s'intersecter, rendant ainsi l'assemblage physiquement impossible.

5.1.4 Nécessité et utilité de tronquer

Il est toujours possible de tronquer les blocs obtenus par cette méthode de construction (pour éviter des angles trop aigus fragilisant la pièce par exemple), mais parfois il est même nécessaire de le faire. La figure Fig. 5.10 représente un assemblage de tétraèdres non réguliers obtenu à partir d'un maillage de carrés (en pointillé sur la figure). Les angles utilisés sont $\alpha_1 = \alpha_3 = 45^\circ$ et $\alpha_2 = \alpha_4 = -30^\circ$ pour les blocs en rouges et $\alpha_1 = \alpha_3 = -45^\circ$ et $\alpha_2 = \alpha_4 = 30^\circ$ pour ceux en bleu. Utiliser des angles non égaux en valeur absolue sur un maillage de carrés entraîne que les arêtes supérieures (resp. inférieures) des tétraèdres bleus et rouges ne sont pas à la même hauteur. Les tétraèdres rouges (resp. bleus) se superposent alors partiellement, d'où la nécessité de tronquer. Le même phénomène est également visible sur la Fig. 5.9c.

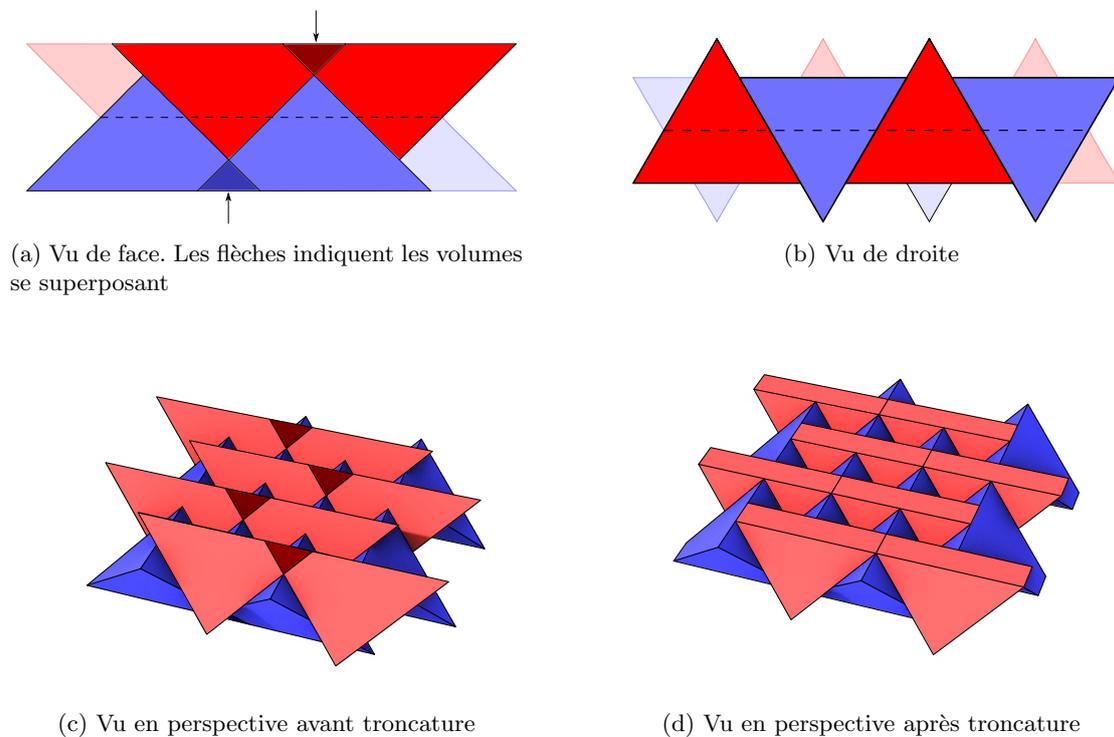
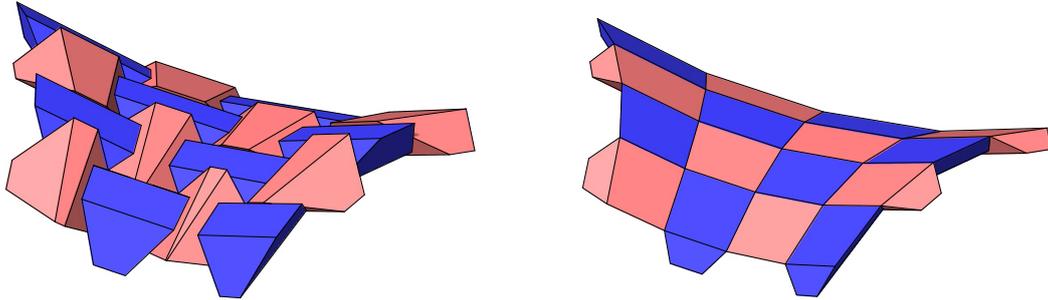
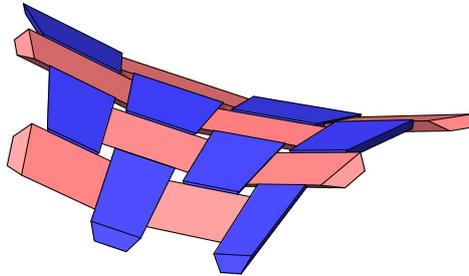


FIGURE 5.10 – Nécessité de tronquer

La troncation a également d'autres utilités. Les tronctures présentées jusqu'à présent ont toujours été effectuées de part et d'autre du maillage (Fig. 5.11a). En effectuant la troncture par le plan P_{h_1} avec $h_1 = 0$ (i.e. par le plan de la face du maillage), on obtient une surface "lisse" (Fig. 5.11b). En effectuant les tronctures par des plans P_{h_1} et P_{h_2} du même côté du maillage (h_1 et h_2 de même signe), on peut créer des ouvertures dans l'assemblage (Fig. 5.11c).



(a) P_{h_1} et P_{h_2} de part et d'autre du maillage ($h_1 > 0$ et $h_2 < 0$) (b) P_{h_1} confondu avec le plan du maillage ($h_1 = 0$ et $h_2 < 0$)



(c) P_{h_1} et P_{h_2} du même côté du maillage ($h_1 < 0$ et $h_2 < 0$) permettant de créer des ouvertures

FIGURE 5.11 – Utilité de la troncature. L'assemblage de la Fig. 5.9c est tronqué par les plans P_{h_1} et P_{h_2} pour différentes valeurs de h_1 et h_2 .

5.2 À propos du blocage

L'autoblocage topologique a été défini en Sec. 2.2.1. Nous rappelons ici les trois définitions retenues :

Définition 1 : Dans un assemblage de blocs, un bloc est bloqué en translation (resp. rotation) s'il n'existe aucun mouvement infinitésimal de translation rectiligne (resp. rotation) possible permettant de l'enlever de l'assemblage en considérant ses voisins comme fixes.

Définition 2 : Un assemblage de N blocs est topologiquement autobloquant en translation (resp. rotation) si chacun de ses N blocs est bloqué en translation (resp. rotation).

Définition 3 : Un assemblage de N blocs est **totalemment** topologiquement autobloquant s'il est topologiquement autobloquant en translation et en rotation.

Dans cette section, nous nous intéressons particulièrement au blocage en translation des blocs et nous exposons des conditions simples pour qu'un bloc soit bloqué en translation. Nous montrons également un résultat à notre connaissance nouveau et contre-intuitif ; à savoir que construire les blocs avec des angles α_i non égaux peut conduire à un assemblage dans lequel chacun des blocs est bloqué en translation, mais un sous-ensemble – constitué de plusieurs de ces blocs – ne l'est pas.

5.2.1 Conditions de blocage en translation

Soient F_1 et F_2 les deux faces de la Fig. 5.5a. Soient V^{F_1} le bloc issu de la face F_1 et des plans $\{P_1^{F_1}, P_2^{F_1}, P_3^{F_1}, P_4^{F_1}\}$ et V^{F_2} le bloc issu de la face F_2 et des plans $\{P_1^{F_2}, P_2^{F_2}, P_3^{F_2}, P_4^{F_2}\}$. Le bloc V^{F_1} est représenté Fig. 5.12. Cette figure permet de remarquer que $P_1^{F_1} = P_3^{F_2}$. Le plan $P_3^{F_2}$ peut être vu de deux manières :

1. $P_3^{F_2}$ est un des quatre plans permettant de construire le bloc V^{F_2} .
2. $P_3^{F_2}$ (égal à l'orientation près au plan $P_1^{F_1}$) correspond à la surface du bloc voisin V^{F_1} . Ce plan empêche donc toute translation du bloc V^{F_2} dans la direction du bloc V^{F_1} .

Dès lors que le bloc n'est pas à la frontière de l'assemblage, cette remarque reste vraie pour chaque plan $P_i^{F_2}$. Ainsi, les quatre voisins à un bloc peuvent ou non empêcher tout mouvement de translation de ce bloc en fonction de l'agencement de ces quatre plans. Nous allons dans la suite de cette section trouver une condition nécessaire et suffisante sur les plans P_1^F, P_2^F, P_3^F et P_4^F afin que le bloc issu de la face F soit bloqué en translation. Autrement dit, que les quatre voisins du bloc considéré empêchent tout mouvement de translation de ce dernier.

Considérons un bloc V (Fig. 5.13) issu des plans P_1, P_2, P_3 et P_4 (par souci de clarté, nous ne mentionnons pas l'indice F). Soit $\vec{u} \in E^*$ un vecteur déplacement de ce bloc. Par construction, le bloc V est du côté positif de chaque plan P_i de normale \vec{n}_i et contenant l'arête de vecteur directeur \vec{t}_i (cf. Fig. 5.3). Ainsi, \vec{u} peut se décomposer sur la base $(\vec{t}_i, \vec{n}_i, \vec{t}_i \wedge \vec{n}_i)$ par :

$$\vec{u} = (\vec{u} \cdot \vec{t}_i) \vec{t}_i + (\vec{u} \cdot \vec{n}_i) \vec{n}_i + \left(\vec{u} \cdot (\vec{t}_i \wedge \vec{n}_i) \right) (\vec{t}_i \wedge \vec{n}_i)$$

Le plan P_i correspond aussi à un des plans de la surface du bloc voisin. Ainsi, si la composante selon \vec{n}_i du vecteur déplacement \vec{u} est négative ($\vec{u} \cdot \vec{n}_i < 0$), le bloc V entre

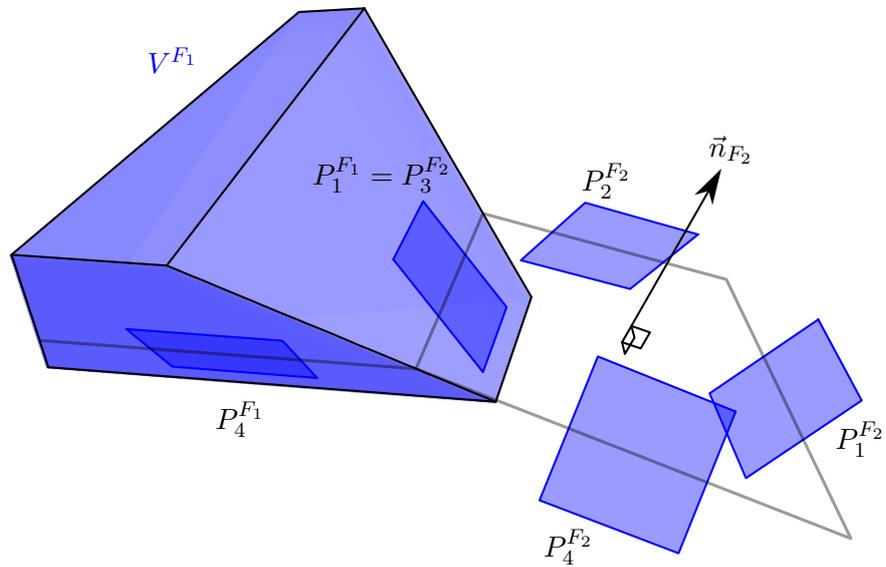
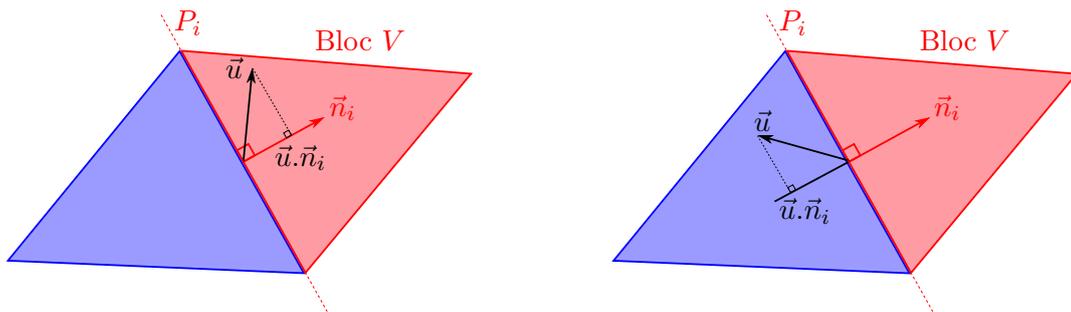


FIGURE 5.12 – Bloc V^{F_1} et plans $\{P_1^{F_2}, P_2^{F_2}, P_3^{F_2}, P_4^{F_2}\}$ permettant de construire le bloc V^{F_2} . À noter, qu'à l'orientation près, $P_1^{F_1} = P_3^{F_2}$.

en collision avec le bloc voisin (Fig. 5.13b). À l'inverse, si la composante est positive ($\vec{u} \cdot \vec{n}_i \geq 0$), le déplacement est possible selon \vec{u} (Fig. 5.13a).

Le plan P_i , de normale \vec{n}_i , empêche donc la translation du bloc V selon \vec{u} si, et seulement si, $\vec{u} \cdot \vec{n}_i < 0$.



(a) Le déplacement du bloc V selon \vec{u} est possible ($\vec{u} \cdot \vec{n}_i > 0$).

(b) Le déplacement du bloc V selon \vec{u} n'est pas possible. Le bloc entre en collision avec son voisin ($\vec{u} \cdot \vec{n}_i < 0$).

FIGURE 5.13

Le déplacement du bloc V selon une translation \vec{u} est impossible si, et seulement si, au moins l'un des quatre plans P_1, P_2, P_3 ou P_4 l'en empêche. On a alors :

$$\boxed{\text{Le bloc est bloqué en translation} \Leftrightarrow \forall \vec{u} \in E^*, \left\{ \begin{array}{l} \vec{u} \cdot \vec{n}_1 < 0 \\ \text{ou} \vec{u} \cdot \vec{n}_2 < 0 \\ \text{ou} \vec{u} \cdot \vec{n}_3 < 0 \\ \text{ou} \vec{u} \cdot \vec{n}_4 < 0 \end{array} \right.} \quad (5.1)$$

De même, le bloc n'est pas bloqué en translation si, et seulement s'il existe un vecteur déplacement \vec{u} qui n'est bloqué par aucun des quatre plans P_1, P_2, P_3 et P_4 :

$$\boxed{\text{Le bloc n'est pas bloqué en translation} \Leftrightarrow \exists \vec{u} \in E^*, \left\{ \begin{array}{l} \vec{u} \cdot \vec{n}_1 \geq 0 \\ \text{et} \vec{u} \cdot \vec{n}_2 \geq 0 \\ \text{et} \vec{u} \cdot \vec{n}_3 \geq 0 \\ \text{et} \vec{u} \cdot \vec{n}_4 \geq 0 \end{array} \right.} \quad (5.2)$$

L'inconvénient de l'équivalence Eq. 5.1 est que pour montrer que le bloc est bloqué en translation, il faut vérifier l'inégalité de droite **pour tout** vecteur \vec{u} . Nous allons désormais démontrer l'équivalence Eq. 5.3 qui permet également de montrer qu'un bloc est bloqué en translation ou non, sans avoir besoin de vérifier une inégalité pour tout vecteur \vec{u} . Notons que cette inégalité ne dépend que des normales $\vec{n}_1, \vec{n}_2, \vec{n}_3$ et \vec{n}_4 et est indépendante du type de quadrangle (plan ou non plan) sur lequel ont été construits les plans P_1, P_2, P_3 et P_4 . Pour la démontrer, nous faisons l'hypothèse supplémentaire $\det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \neq 0$.

$$\boxed{\text{Le bloc est bloqué en translation} \Leftrightarrow \left\{ \begin{array}{l} \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) < 0 \quad \text{(i)} \\ \text{et} \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_2, \vec{n}_3, \vec{n}_4) < 0 \quad \text{(ii)} \\ \text{et} \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_3, \vec{n}_1, \vec{n}_4) < 0 \quad \text{(iii)} \end{array} \right.} \quad (5.3)$$

Démonstration

Nous démontrons ici l'équivalence Eq. 5.3 en démontrant l'implication directe et sa réciproque.

\Rightarrow Pour démontrer la première implication, supposons que le bloc est bloqué en translation. Autrement dit, selon Eq. 5.1, pour tout \vec{u} dans E^* , il existe $i \in \{1, 2, 3, 4\}$ tel que $\vec{u} \cdot \vec{n}_i < 0$. Nous souhaitons alors montrer que les inégalités (i), (ii) et (iii) de Eq. 5.3 sont vérifiées.

Par hypothèse, pour tout $\vec{u} \in E^*$, il existe $i \in \{1, 2, 3, 4\}$ tel que $\vec{u} \cdot \vec{n}_i < 0$. Cette inégalité étant valable pour tout \vec{u} , en choisissant bien le vecteur \vec{u} , nous pouvons montrer facilement les inégalités voulues.

En effet, posons $\vec{u} = \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot (\vec{n}_1 \wedge \vec{n}_2) \in E^*$. Ce vecteur correspond en fait à un des vecteurs de la base représentée en rouge Fig. 5.14. On a alors ² :

$$\begin{cases} \vec{u} \cdot \vec{n}_1 = 0 \\ \vec{u} \cdot \vec{n}_2 = 0 \\ \vec{u} \cdot \vec{n}_3 = (\det(\vec{n}_1, \vec{n}_2, \vec{n}_3))^2 \\ \vec{u} \cdot \vec{n}_4 = \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) \end{cases} \geq 0$$

Par hypothèse, au moins l'une de ces valeurs est strictement inférieure à 0. Les trois premières valeurs étant soit nulle, soit positive, on a donc nécessairement $\vec{u} \cdot \vec{n}_4 < 0$. Autrement dit, on retrouve l'inégalité (i) de l'Eq.5.3 :

$$\det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) < 0$$

De manière similaire, avec $\vec{v} = \det(\vec{n}_2, \vec{n}_3, \vec{n}_1) \cdot (\vec{n}_2 \wedge \vec{n}_3) \in E^*$, on a :

$$\begin{cases} \vec{v} \cdot \vec{n}_1 = (\det(\vec{n}_2, \vec{n}_3, \vec{n}_1))^2 \\ \vec{v} \cdot \vec{n}_2 = 0 \\ \vec{v} \cdot \vec{n}_3 = 0 \\ \vec{v} \cdot \vec{n}_4 = \det(\vec{n}_2, \vec{n}_3, \vec{n}_1) \cdot \det(\vec{n}_2, \vec{n}_3, \vec{n}_4) \end{cases} \geq 0$$

On a nécessairement $\vec{v} \cdot \vec{n}_4 < 0$, qui n'est autre que l'inégalité (ii) de l'Eq.5.3. Enfin, on prouve l'inégalité (iii) de l'Eq.5.3 avec $\vec{w} = \det(\vec{n}_3, \vec{n}_1, \vec{n}_2) \cdot (\vec{n}_3 \wedge \vec{n}_1) \in E^*$.

$\boxed{\Leftarrow}$ Démontrons désormais l'implication réciproque. Supposons ainsi que les inégalités (i), (ii) et (iii) de l'Eq.5.3 sont respectées et montrons que le bloc est nécessairement bloqué en translation. Pour cela, procédons par l'absurde, en montrant que si le bloc n'est pas bloqué en translation, alors on arrive à une contradiction.

Notons $\epsilon = \det(\vec{n}_1, \vec{n}_2, \vec{n}_3)$. D'après l'inégalité (i), $\epsilon \neq 0$, on a alors :

$$\det(\epsilon \cdot \vec{n}_1 \wedge \vec{n}_2, \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3, \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1) \neq 0$$

En effet, en exprimant le déterminant comme un produit mixte, on obtient :

$$\begin{aligned} \det(\epsilon \cdot \vec{n}_1 \wedge \vec{n}_2, \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3, \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1) &= \epsilon^3 \cdot [(\vec{n}_1 \wedge \vec{n}_2) \wedge (\vec{n}_2 \wedge \vec{n}_3)] \cdot (\vec{n}_3 \wedge \vec{n}_1) \\ &= \epsilon^3 \cdot [((\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_3) \vec{n}_2 - ((\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_2) \vec{n}_3] \cdot (\vec{n}_3 \wedge \vec{n}_1) \\ &= \epsilon^3 \cdot [\det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \vec{n}_2 - 0 \vec{n}_3] \cdot (\vec{n}_3 \wedge \vec{n}_1) \\ &= \epsilon^3 \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) (\vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_2 \\ &= \epsilon^3 \cdot (\det(\vec{n}_1, \vec{n}_2, \vec{n}_3))^2 \\ &= \epsilon^5 \neq 0 \end{aligned}$$

2. Dans ce chapitre, l'expression du déterminant en tant que produite mixte est régulièrement utilisé. Nous rappelons que $\det(\vec{a}, \vec{b}, \vec{c}) = (\vec{a} \wedge \vec{b}) \cdot \vec{c}$. De plus, le déterminant est invariant par toute permutation circulaire, c'est-à-dire, $(\vec{a} \wedge \vec{b}) \cdot \vec{c} = (\vec{b} \wedge \vec{c}) \cdot \vec{a} = (\vec{c} \wedge \vec{a}) \cdot \vec{b}$.

La famille $\mathfrak{B} = (\epsilon \cdot \vec{n}_1 \wedge \vec{n}_2, \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3, \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1)$ est donc libre et de cardinal 3, c'est donc une base de E . Cette base a une interprétation géométrique simple, représentée sur la Fig. 5.14. Elle est composée des vecteurs directeurs des intersections des plans deux à deux. Le scalaire ϵ permettant que ces vecteurs "pointent" du bon côté. Plus de détails sont donnés sur cette base à la fin de cette démonstration.

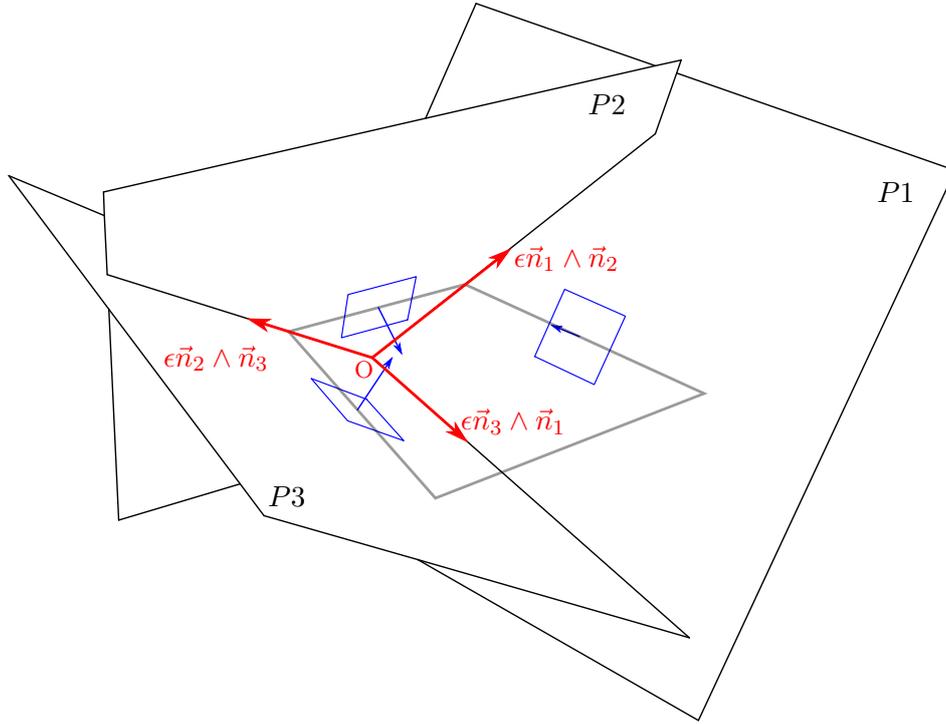


FIGURE 5.14 – Représentation de la base \mathfrak{B} avec $\epsilon = \det(\vec{n}_1, \vec{n}_2, \vec{n}_3)$

Par hypothèse, le bloc n'est pas bloqué en translation, alors, d'après Eq. 5.2, il existe $\vec{u}_0 \in E^*$ tel que $\vec{u}_0 \cdot \vec{n}_i \geq 0$ pour tout $i \in \{1, 2, 3, 4\}$. Exprimons \vec{u}_0 dans la base \mathfrak{B} :

Soient $A, B, C \in \mathbb{R}$, non tous nuls, tels que

$$\vec{u}_0 = A \cdot \epsilon \cdot \vec{n}_1 \wedge \vec{n}_2 + B \cdot \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3 + C \cdot \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1$$

On a alors :

$$\begin{cases} \vec{u}_0 \cdot \vec{n}_1 = B \cdot \epsilon \cdot (\vec{n}_2 \wedge \vec{n}_3) \cdot \vec{n}_1 = B \cdot \epsilon \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) = B \cdot \epsilon^2 \\ \vec{u}_0 \cdot \vec{n}_2 = C \cdot \epsilon \cdot (\vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_2 = C \cdot \epsilon \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) = C \cdot \epsilon^2 \\ \vec{u}_0 \cdot \vec{n}_3 = A \cdot \epsilon \cdot (\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_3 = A \cdot \epsilon \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) = A \cdot \epsilon^2 \\ \vec{u}_0 \cdot \vec{n}_4 = A \cdot \epsilon \cdot (\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_4 + B \cdot \epsilon \cdot (\vec{n}_2 \wedge \vec{n}_3) \cdot \vec{n}_4 + C \cdot \epsilon \cdot (\vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_4 \end{cases}$$

Or, pour tout $i \in \{1, 2, 3, 4\}$, $\vec{u}_0 \cdot \vec{n}_i \geq 0$. Autrement dit :

$$\left\{ \begin{array}{l} \vec{u}_0 \cdot \vec{n}_1 \geq 0 \Leftrightarrow B \cdot \epsilon^2 \geq 0 \\ \quad \Leftrightarrow B \geq 0 \\ \\ \vec{u}_0 \cdot \vec{n}_2 \geq 0 \Leftrightarrow C \cdot \epsilon^2 \geq 0 \\ \quad \Leftrightarrow C \geq 0 \\ \\ \vec{u}_0 \cdot \vec{n}_3 \geq 0 \Leftrightarrow A \cdot \epsilon^2 \geq 0 \\ \quad \Leftrightarrow A \geq 0 \\ \\ \vec{u}_0 \cdot \vec{n}_4 \geq 0 \Leftrightarrow A \cdot \epsilon \cdot (\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_4 + B \cdot \epsilon \cdot (\vec{n}_2 \wedge \vec{n}_3) \cdot \vec{n}_4 + C \cdot \epsilon \cdot (\vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_4 \geq 0 \end{array} \right.$$

Ainsi, $A \geq 0$, $B \geq 0$ et $C \geq 0$. Écrivons d'une autre manière la quatrième inégalité ($\vec{u}_0 \cdot \vec{n}_4 \geq 0$) :

$$\begin{aligned} \vec{u}_0 \cdot \vec{n}_4 &= A \cdot \epsilon \cdot (\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_4 + B \cdot \epsilon \cdot (\vec{n}_2 \wedge \vec{n}_3) \cdot \vec{n}_4 + C \cdot \epsilon \cdot (\vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_4 \\ &= A \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) \\ &\quad + B \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_2, \vec{n}_3, \vec{n}_4) \\ &\quad + C \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_3, \vec{n}_1, \vec{n}_4) \\ &= T_1 + T_2 + T_3 \end{aligned}$$

avec

$$\begin{aligned} T_1 &= A \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) \\ T_2 &= B \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_2, \vec{n}_3, \vec{n}_4) \\ T_3 &= C \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_3, \vec{n}_1, \vec{n}_4) \end{aligned}$$

On a alors

$$\vec{u}_0 \cdot \vec{n}_4 \geq 0 \Leftrightarrow T_1 + T_2 + T_3 \geq 0 \quad (5.4)$$

Puisque $A \geq 0$, $B \geq 0$ et $C \geq 0$, les inégalités (i), (ii) et (iii) de l'Eq. 5.3 entraînent alors que $T_1 \leq 0$, $T_2 \leq 0$ et $T_3 \leq 0$. Alors $T_1 + T_2 + T_3 \leq 0$. L'inégalité Eq. 5.4 entraîne alors :

$$T_1 + T_2 + T_3 = 0$$

T_1 , T_2 et T_3 étant de même signe, on a alors nécessairement :

$$\begin{aligned} T_1 &= 0 \\ T_2 &= 0 \\ T_3 &= 0 \end{aligned}$$

T_1 est le produit de $A \geq 0$ et d'une quantité strictement négative (d'après l'inégalité (i) de l'Eq. 5.3). On a donc $A = 0$. De même, pour T_2 et T_3 on obtient $B = 0$ et $C = 0$.
Or

$$\vec{u}_0 = A \cdot \epsilon \cdot \vec{n}_1 \wedge \vec{n}_2 + B \cdot \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3 + C \cdot \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1$$

Donc $\vec{u}_0 = 0$, ce qui est impossible car par hypothèse \vec{u}_0 ne peut pas être le vecteur nul. En ayant supposé que le bloc n'est pas bloqué en translation, on est arrivé à une contradiction. Le bloc est donc nécessairement bloqué en translation. Ce qui conclut cette démonstration.

Remarque Revenons sur la base $\mathfrak{B} = (\epsilon \cdot \vec{n}_1 \wedge \vec{n}_2, \epsilon \cdot \vec{n}_2 \wedge \vec{n}_3, \epsilon \cdot \vec{n}_3 \wedge \vec{n}_1)$ introduite dans cette démonstration. Cette base est représentée Fig. 5.14. Supposons $\epsilon = \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \neq 0$. Les plans P_1 , P_2 et P_3 sont donc non parallèles deux à deux et s'intersectent en un point que l'on note O . Pour $i \neq j$, le vecteur $\vec{n}_i \wedge \vec{n}_j$ est un vecteur directeur de la droite d'intersection des plans P_i et P_j . Le vecteur $\epsilon \cdot \vec{n}_i \wedge \vec{n}_j$ l'est donc aussi. De plus, pour tout $k \in \{1, 2, 3\}$,

$$\begin{cases} (\epsilon \cdot \vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_k = \epsilon \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_k) \geq 0 \\ (\epsilon \cdot \vec{n}_2 \wedge \vec{n}_3) \cdot \vec{n}_k = \epsilon \cdot \det(\vec{n}_2, \vec{n}_3, \vec{n}_k) \geq 0 \\ (\epsilon \cdot \vec{n}_3 \wedge \vec{n}_1) \cdot \vec{n}_k = \epsilon \cdot \det(\vec{n}_3, \vec{n}_1, \vec{n}_k) \geq 0 \end{cases}$$

Ces inégalités montrent que chacun des trois vecteurs de la base \mathfrak{B} pointe du côté positif de chacun des plans P_1 , P_2 et P_3 .

5.2.2 Cas particuliers des quadrangles plans

5.2.2.1 Quadrangle plan quelconque

L'équivalence 5.3 ne dépend que des normales \vec{n}_1 , \vec{n}_2 , \vec{n}_3 et \vec{n}_4 et est indépendante du type de quadrangle (plan ou non plan) sur lequel ont été construits les plans P_1 , P_2 , P_3 et P_4 . En considérant désormais les quadrangles comme plans et en utilisant les notations introduites en Sec. 5.1.2, nous pouvons réécrire les conditions (i), (ii) et (iii) de l'Eq. 5.3 en fonction de $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ et des vecteurs directeurs des arêtes du quadrangle $\{\vec{t}_1, \vec{t}_2, \vec{t}_3, \vec{t}_4\}$ (Fig. 5.15). Ce qui donne l'équivalence de l'Eq. 5.5.

Le bloc, construit à partir d'un quadrangle plan, est bloqué en translation

\Updownarrow

$$\left\{ \begin{array}{l}
 \left[\tan \alpha_3 \sin \widehat{(\vec{t}_1, \vec{t}_2)} + \tan \alpha_2 \sin \widehat{(\vec{t}_3, \vec{t}_1)} + \tan \alpha_1 \sin \widehat{(\vec{t}_2, \vec{t}_3)} \right] \\
 \quad \cdot \left[\tan \alpha_4 \sin \widehat{(\vec{t}_1, \vec{t}_2)} + \tan \alpha_2 \sin \widehat{(\vec{t}_4, \vec{t}_1)} + \tan \alpha_1 \sin \widehat{(\vec{t}_2, \vec{t}_4)} \right] < 0 \\
 \text{et} \left[\tan \alpha_3 \sin \widehat{(\vec{t}_1, \vec{t}_2)} + \tan \alpha_2 \sin \widehat{(\vec{t}_3, \vec{t}_1)} + \tan \alpha_1 \sin \widehat{(\vec{t}_2, \vec{t}_3)} \right] \\
 \quad \cdot \left[\tan \alpha_3 \sin \widehat{(\vec{t}_4, \vec{t}_2)} + \tan \alpha_2 \sin \widehat{(\vec{t}_3, \vec{t}_4)} + \tan \alpha_4 \sin \widehat{(\vec{t}_2, \vec{t}_3)} \right] < 0 \\
 \text{et} \left[\tan \alpha_3 \sin \widehat{(\vec{t}_1, \vec{t}_2)} + \tan \alpha_2 \sin \widehat{(\vec{t}_3, \vec{t}_1)} + \tan \alpha_1 \sin \widehat{(\vec{t}_2, \vec{t}_3)} \right] \\
 \quad \cdot \left[\tan \alpha_1 \sin \widehat{(\vec{t}_4, \vec{t}_3)} + \tan \alpha_3 \sin \widehat{(\vec{t}_1, \vec{t}_4)} + \tan \alpha_4 \sin \widehat{(\vec{t}_3, \vec{t}_1)} \right] < 0
 \end{array} \right. \quad (5.5)$$

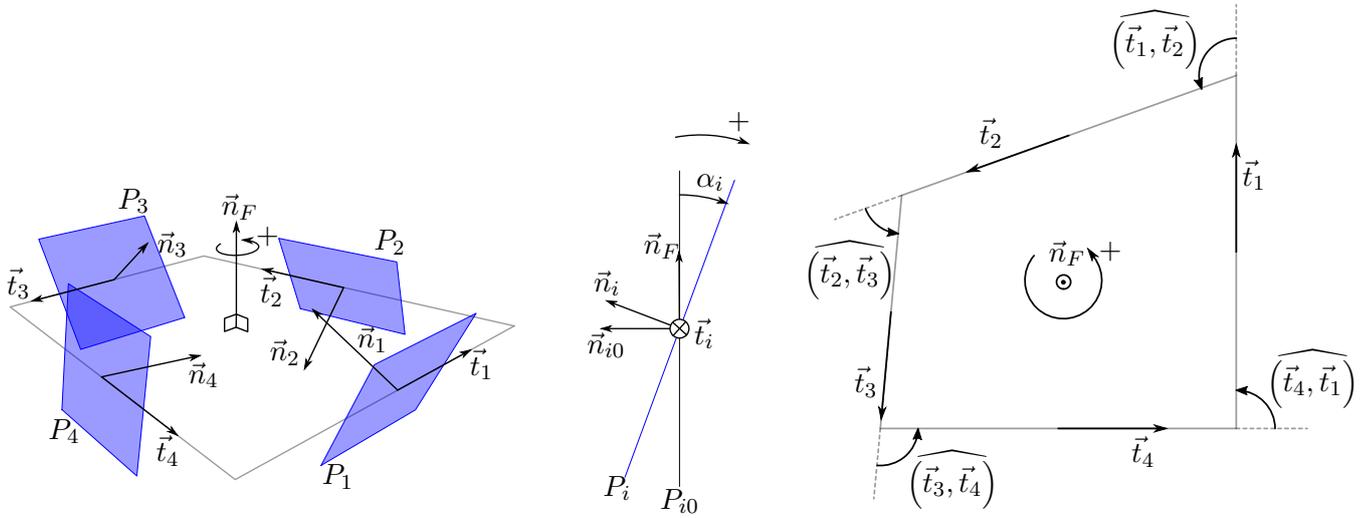


FIGURE 5.15 – Rappel des notations

Démonstration

Pour tout $i \in \{1, 2, 3, 4\}$, on a $\vec{n}_i = \cos \alpha_i \vec{n}_{i0} + \sin \alpha_i \vec{n}_F$ (Fig. 5.15). Ainsi,

$$\begin{aligned} \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) &= (\vec{n}_1 \wedge \vec{n}_2) \cdot \vec{n}_3 \\ &= [(\cos \alpha_1 \vec{n}_{10} + \sin \alpha_1 \vec{n}_F) \wedge (\cos \alpha_2 \vec{n}_{20} + \sin \alpha_2 \vec{n}_F)] \cdot \vec{n}_3 \\ &= [\cos \alpha_1 \cos \alpha_2 (\vec{n}_{10} \wedge \vec{n}_{20}) \\ &\quad + \cos \alpha_1 \sin \alpha_2 (\vec{n}_{10} \wedge \vec{n}_F) \\ &\quad + \sin \alpha_1 \cos \alpha_2 (\vec{n}_F \wedge \vec{n}_{20})] \cdot \vec{n}_3 \end{aligned}$$

Or $(\vec{n}_F, \vec{t}_i, \vec{n}_{i0})$ est un trièdre direct, c'est-à-dire :

$$\vec{n}_{i0} = \vec{n}_F \wedge \vec{t}_i$$

On a alors :

$$\begin{aligned} \vec{n}_{10} \wedge \vec{n}_{20} &= (\vec{n}_F \wedge \vec{t}_1) \wedge (\vec{n}_F \wedge \vec{t}_2) \\ &= [(\vec{n}_F \wedge \vec{t}_1) \cdot \vec{t}_2] \vec{n}_F - [(\vec{n}_F \wedge \vec{t}_1) \cdot \vec{n}_F] \vec{t}_2 \quad (\text{double produit vectoriel}) \\ &= [(\vec{n}_F \wedge \vec{t}_1) \cdot \vec{t}_2] \vec{n}_F \\ &= [(\vec{t}_1 \wedge \vec{t}_2) \cdot \vec{n}_F] \vec{n}_F \\ &= \sin(\widehat{\vec{t}_1, \vec{t}_2}) \vec{n}_F \end{aligned}$$

En utilisant cette dernière égalité dans l'expression trouvée précédemment, on a alors :

$$\begin{aligned} \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) &= \left[\cos \alpha_1 \cos \alpha_2 \sin(\widehat{\vec{t}_1, \vec{t}_2}) \vec{n}_F \right. \\ &\quad \left. + \cos \alpha_1 \sin \alpha_2 (\vec{n}_{10} \wedge \vec{n}_F) \right. \\ &\quad \left. + \sin \alpha_1 \cos \alpha_2 (\vec{n}_F \wedge \vec{n}_{20}) \right] \cdot \vec{n}_3 \\ &= \left[\cos \alpha_1 \cos \alpha_2 \sin(\widehat{\vec{t}_1, \vec{t}_2}) \vec{n}_F \right. \\ &\quad \left. + \cos \alpha_1 \sin \alpha_2 \vec{t}_1 \right. \\ &\quad \left. + \sin \alpha_1 \cos \alpha_2 (-\vec{t}_2) \right] \cdot \vec{n}_3 \end{aligned}$$

Or

$$\begin{aligned}\vec{n}_F \cdot \vec{n}_3 &= \left[\sin \alpha_3 \vec{n}_3 + \cos \alpha_3 (\vec{t}_3 \wedge \vec{n}_3) \right] \cdot \vec{n}_3 \\ &= \sin \alpha_3\end{aligned}$$

$$\begin{aligned}\vec{t}_1 \cdot \vec{n}_3 &= \vec{t}_1 \cdot [\cos \alpha_3 \vec{n}_{30} + \sin \alpha_3 \vec{n}_F] \\ &= \cos \alpha_3 (\vec{n}_{30} \cdot \vec{t}_1) \\ &= \cos \alpha_3 (\vec{n}_F \wedge \vec{t}_3) \cdot \vec{t}_1 \\ &= \cos \alpha_3 (\vec{t}_3 \wedge \vec{t}_1) \cdot \vec{n}_F \\ &= \cos \alpha_3 \sin (\widehat{\vec{t}_3, \vec{t}_1})\end{aligned}$$

$$\begin{aligned}-\vec{t}_2 \cdot \vec{n}_3 &= -\vec{t}_2 \cdot [\cos \alpha_3 \vec{n}_{30} + \sin \alpha_3 \vec{n}_F] \\ &= \cos \alpha_3 (\vec{n}_{30} \cdot (-\vec{t}_2)) \\ &= \cos \alpha_3 (\vec{n}_F \wedge \vec{t}_3) \cdot (-\vec{t}_2) \\ &= \cos \alpha_3 (\vec{t}_3 \wedge (-\vec{t}_2)) \cdot \vec{n}_F \\ &= \cos \alpha_3 (\vec{t}_2 \wedge \vec{t}_3) \cdot \vec{n}_F \\ &= \cos \alpha_3 \sin (\widehat{\vec{t}_2, \vec{t}_3})\end{aligned}$$

On obtient alors :

$$\begin{aligned}\det(\vec{n}_1, \vec{n}_2, \vec{n}_3) &= \cos \alpha_1 \cos \alpha_2 \sin \alpha_3 \sin (\widehat{\vec{t}_1, \vec{t}_2}) \\ &\quad + \cos \alpha_1 \sin \alpha_2 \cos \alpha_3 \sin (\widehat{\vec{t}_3, \vec{t}_1}) \\ &\quad + \sin \alpha_1 \cos \alpha_2 \cos \alpha_3 \sin (\widehat{\vec{t}_2, \vec{t}_3})\end{aligned}$$

De manière similaire, on obtient :

$$\begin{aligned}\det(\vec{n}_1, \vec{n}_2, \vec{n}_4) &= \cos \alpha_1 \cos \alpha_2 \sin \alpha_4 \sin (\widehat{\vec{t}_1, \vec{t}_2}) \\ &\quad + \cos \alpha_1 \sin \alpha_2 \cos \alpha_4 \sin (\widehat{\vec{t}_4, \vec{t}_1}) \\ &\quad + \sin \alpha_1 \cos \alpha_2 \cos \alpha_4 \sin (\widehat{\vec{t}_2, \vec{t}_4})\end{aligned}$$

Puisque $-90^\circ < \alpha_i < 90^\circ$, alors $\cos \alpha_i > 0$. Diviser $\det(\vec{n}_1, \vec{n}_2, \vec{n}_3)$ par $\cos \alpha_1 \cos \alpha_2 \cos \alpha_3$ ne change donc pas son signe. De même en divisant $\det(\vec{n}_1, \vec{n}_2, \vec{n}_4)$ par $\cos \alpha_1 \cos \alpha_2 \cos \alpha_4$. L'inégalité (i) de l'Eq. 5.3 peut alors se réécrire :

$$\det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) < 0 \Leftrightarrow \frac{\det(\vec{n}_1, \vec{n}_2, \vec{n}_3)}{\cos \alpha_1 \cos \alpha_2 \cos \alpha_3} \cdot \frac{\det(\vec{n}_1, \vec{n}_2, \vec{n}_4)}{\cos \alpha_1 \cos \alpha_2 \cos \alpha_4} < 0$$

$$\boxed{\begin{aligned} & \det(\vec{n}_1, \vec{n}_2, \vec{n}_3) \cdot \det(\vec{n}_1, \vec{n}_2, \vec{n}_4) < 0 \\ & \Downarrow \\ & \left[\tan \alpha_3 \sin(\widehat{\vec{t}_1, \vec{t}_2}) + \tan \alpha_2 \sin(\widehat{\vec{t}_3, \vec{t}_1}) + \tan \alpha_1 \sin(\widehat{\vec{t}_2, \vec{t}_3}) \right] \\ & \quad \cdot \left[\tan \alpha_4 \sin(\widehat{\vec{t}_1, \vec{t}_2}) + \tan \alpha_2 \sin(\widehat{\vec{t}_4, \vec{t}_1}) + \tan \alpha_1 \sin(\widehat{\vec{t}_2, \vec{t}_4}) \right] < 0 \end{aligned}}$$

Des résultats similaires sont trouvés en suivant la même démarche pour les inégalités (ii) et (iii) de l'Eq. 5.3. Ce qui conduit à l'équivalence de l'Eq. 5.5 et conclut cette démonstration.

5.2.2.2 Quadrangle parallèle

Si le quadrangle considéré est un parallélogramme (Fig. 5.16), l'équivalence Eq. 5.5 se simplifie.

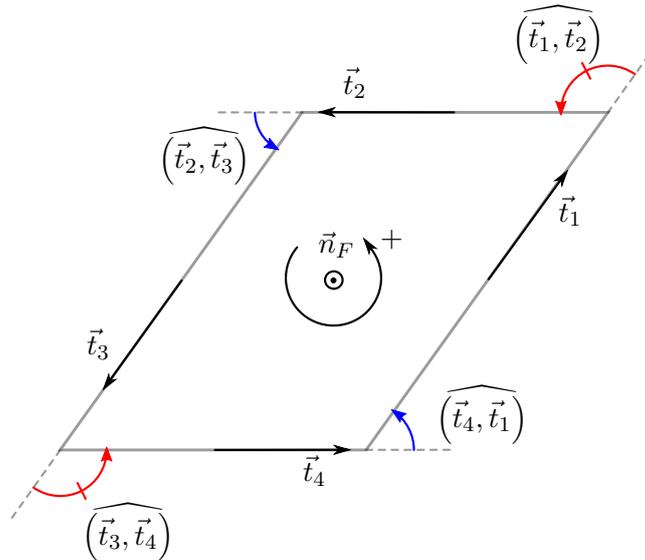


FIGURE 5.16 – Quadrangle parallélogramme

En effet, les angles opposés dans un parallélogramme ont la même valeur :

$$\begin{aligned}\widehat{(\vec{t}_1, \vec{t}_2)} &= \widehat{(\vec{t}_3, \vec{t}_4)} \\ \widehat{(\vec{t}_2, \vec{t}_3)} &= \widehat{(\vec{t}_4, \vec{t}_1)}\end{aligned}$$

De plus, les angles consécutifs dans un parallélogramme sont supplémentaires :

$$\begin{aligned}\widehat{(\vec{t}_1, \vec{t}_2)} + \widehat{(\vec{t}_2, \vec{t}_3)} &= \Pi \\ \widehat{(\vec{t}_3, \vec{t}_4)} + \widehat{(\vec{t}_4, \vec{t}_1)} &= \Pi\end{aligned}$$

Ainsi, nous avons les relations suivantes entre les sinus :

$$\begin{cases} \sin \widehat{(\vec{t}_1, \vec{t}_2)} = \sin \widehat{(\vec{t}_2, \vec{t}_3)} = \sin \widehat{(\vec{t}_3, \vec{t}_4)} = \sin \widehat{(\vec{t}_4, \vec{t}_1)} \\ \sin \widehat{(\vec{t}_1, \vec{t}_3)} = \sin \widehat{(\vec{t}_2, \vec{t}_4)} = 0 \end{cases}$$

Le deuxième membre de l'équivalence Eq. 5.5 devient alors :

$$\begin{cases} \left(\sin \widehat{(\vec{t}_1, \vec{t}_2)} \right)^2 (\tan \alpha_3 + \tan \alpha_1) (\tan \alpha_4 + \tan \alpha_2) < 0 \\ \text{et } \left(\sin \widehat{(\vec{t}_1, \vec{t}_2)} \right)^2 (\tan \alpha_3 + \tan \alpha_1) (\tan \alpha_2 + \tan \alpha_4) < 0 \\ \text{et } \left(\sin \widehat{(\vec{t}_1, \vec{t}_2)} \right)^2 (\tan \alpha_3 + \tan \alpha_1) (-\tan \alpha_1 - \tan \alpha_3) < 0 \end{cases}$$

En multipliant par $\cos \alpha_1 \cos \alpha_2 \cos \alpha_3 \cos \alpha_4$ qui est positif, on ne change pas le signe de l'inégalité et en utilisant la formule de trigonométrie :

$$(\tan A + \tan B) \cdot \cos A \cdot \cos B = \sin(A + B)$$

on obtient l'équivalence suivante valable pour les parallélogrammes :

$$\begin{aligned} \text{Le bloc construit sur un parallélogramme est bloqué en translation} \\ \Downarrow \\ \sin(\alpha_1 + \alpha_3) \cdot \sin(\alpha_2 + \alpha_4) < 0 \end{aligned}$$

(5.6)

5.2.2.3 Applications

L'assemblage de la Fig. 5.1 est obtenu à partir d'un maillage de carrés avec des angles de signes alternés et de valeur $\pm\alpha$. Le carré étant un cas particulier de parallélogramme, la formule de l'Eq. 5.6 s'applique :

$$\sin(\alpha_1 + \alpha_3) \cdot \sin(\alpha_2 + \alpha_4) = -(\sin 2\alpha)^2 < 0$$

Le bloc est donc bloqué en translation. Chaque bloc de l'assemblage étant identique, l'assemblage est alors topologiquement autobloquant en translation.

Plus généralement, pour un bloc construit à partir d'un parallélogramme, si les angles sont non nuls et successivement de signes alternés ($\alpha_i \cdot \alpha_{i+1} < 0$ pour $i \in \mathbb{Z}/4\mathbb{Z}$), l'inégalité 5.6 devient :

$$\sin(\alpha_1 + \alpha_3) \cdot \sin(\alpha_2 + \alpha_4) = -\sin(|\alpha_1| + |\alpha_3|) \cdot \sin(|\alpha_2| + |\alpha_4|) < 0$$

car $0 < |\alpha_i| < 90^\circ$. **Un bloc construit sur un parallélogramme avec des angles α_i de signes alternés est donc bloqué en translation.**

La procédure *moving cross-section* (Sec. 5.1.1) se base sur des pavages réguliers du plan (notamment des pavages de carrés) en alternant les angles. La remarque précédente montre que les assemblages générés sont en effet topologiquement autobloquants. Alternier la valeurs des angles α_i est donc suffisant pour générer des assemblages topologiquement autobloquants mais l'Eq. 5.6 montre que ce n'est pas forcément nécessaire. En effet, la figure 5.17 représente par exemple un assemblage topologiquement autobloquant construit sur un maillage de carrés, avec $\alpha_1 = 60^\circ$, $\alpha_2 = 30^\circ$, $\alpha_3 = -30^\circ$ et $\alpha_4 = -60^\circ$. L'inégalité de l'Eq. 5.6 est respectée alors que les angles consécutifs ne sont pas de signes alternés.

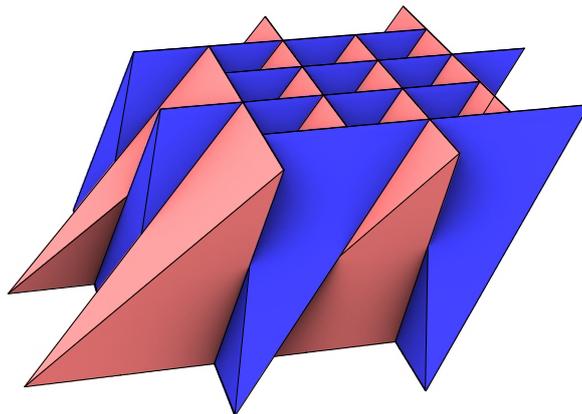


FIGURE 5.17 – Assemblage topologiquement autobloquant de tétraèdres non réguliers. Les angles des plans ne sont pas successivement alternés.

Quadrangle quelconque À l'inverse, en utilisant des angles α_i consécutifs de signes alternés sur un maillage de quadrangles plans non parallélogrammes, le blocage n'est pas forcément assuré. La figure 5.18 montre deux assemblages. L'un construit sur un maillage de carrés, l'autre sur un maillage dont la face centrale est un trapèze. Dans les deux cas, les mêmes angles $\alpha_1 = -15^\circ$, $\alpha_2 = 45^\circ$, $\alpha_3 = -15^\circ$ et $\alpha_4 = 45^\circ$ sont utilisés pour

l'inclinaison des plans P_1 , P_2 , P_3 et P_4 servant à construire le bloc central. Ces angles sont successivement de signes alternés. D'après la remarque précédente, l'assemblage obtenu à partir du maillage de carrés ne contient que des blocs bloqués en translation. En revanche, dans le deuxième assemblage, le bloc issu du trapèze n'est pas bloqué. En effet, pour ce quadrangle particulier :

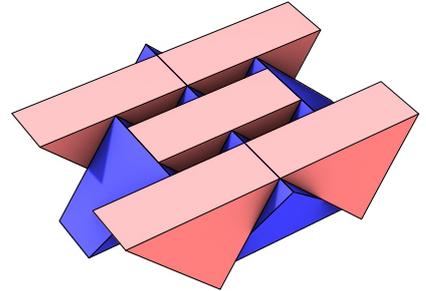
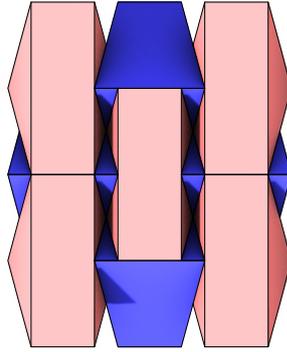
$$\begin{aligned} \widehat{(\vec{t}_1, \vec{t}_2)} &= 110^\circ \\ \widehat{(\vec{t}_2, \vec{t}_3)} &= 110^\circ \\ \widehat{(\vec{t}_3, \vec{t}_4)} &= 70^\circ \\ \widehat{(\vec{t}_4, \vec{t}_1)} &= 70^\circ \end{aligned}$$

Les membres de gauche des inégalités de l'Eq. 5.5, valable pour les quadrangles plans quelconques, deviennent alors :

$$\left\{ \begin{array}{l} [\tan(-15^\circ) \sin(110^\circ) + \tan(45^\circ) \sin(140^\circ) + \tan(-15^\circ) \sin(110^\circ)] \\ \quad \cdot [\tan(45^\circ) \sin(110^\circ) + \tan(45^\circ) \sin(70^\circ) + \tan(-15^\circ) \sin(180^\circ)] = 0.26 \\ \text{et} \quad [\tan(-15^\circ) \sin(110^\circ) + \tan(45^\circ) \sin(140^\circ) + \tan(-15^\circ) \sin(110^\circ)] \\ \quad \cdot [\tan(-15^\circ) \sin(180^\circ) + \tan(45^\circ) \sin(70^\circ) + \tan(45^\circ) \sin(110^\circ)] = 0.26 \\ \text{et} \quad [\tan(-15^\circ) \sin(110^\circ) + \tan(45^\circ) \sin(140^\circ) + \tan(-15^\circ) \sin(110^\circ)] \\ \quad \cdot [\tan(-15^\circ) \sin(110^\circ) + \tan(-15^\circ) \sin(290^\circ) + \tan(45^\circ) \sin(140^\circ)] = 0.089 \end{array} \right.$$

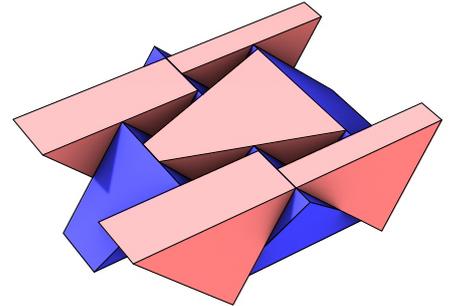
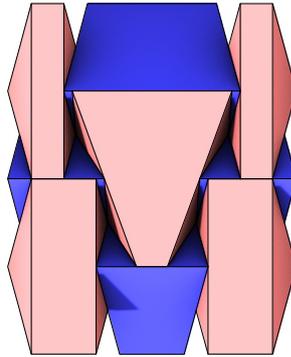
Aucun des membres de ce système n'est négatif, d'après Eq. 5.5, le bloc issu de ce quadrangle n'est donc pas bloqué en translation. Il existe au moins une translation possible qui permet de sortir le bloc de l'assemblage sans collision avec ses voisins. Une telle translation est représentée Fig. 5.18c.

+45°	-45°	+45°
-15°	-15°+15°	+15°-15°
+45°	-45°	+45°
-45°	+45°	-45°
+15°	+15°-15°	-15°+15°
+15°		+15°
-45°	+45°	-45°
+45°	-45°	+45°
-15°	-15°+15°	+15°-15°
-15°		-15°
+45°	-45°	+45°

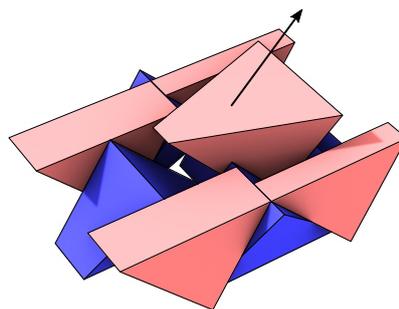


(a) Assemblage sur un pavage régulier de carrés. Les valeurs indiquées sur les arêtes de la figure de gauche correspondent aux valeurs des angles α_i . Pour chaque bloc, les valeurs de α_1 , α_2 , α_3 et α_4 sont successivement de signes alternés. L'assemblage est topologiquement autobloquant.

+45°	-45°	+45°
-15°	-15°+15°	+15°-15°
+45°	-45°	+45°
-45°	+45°	-45°
+15°	+15°-15°	-15°+15°
+15°		+15°
-45°	+45°	-45°
+45°	-45°	+45°
-15°	-15°+15°	+15°-15°
-15°		-15°
+45°	-45°	+45°



(b) Assemblage sur un pavage non régulier. Le quadrangle central est trapézoïdal. Les valeurs des angles α_i sont identiques à celles de (a) mais l'assemblage n'est pas topologiquement autobloquant.



(c) Le bloc issu du quadrangle trapézoïdal n'est pas bloqué en translation. La flèche noire indique un mouvement de translation permettant de le sortir sans en être empêché par les blocs voisins.

FIGURE 5.18 – Assemblages sur des pavages du plans différents mais avec les mêmes angles α_i (de signes alternés pour chaque bloc). Le premier assemblage est topologiquement autobloquant mais pas le second.

5.2.3 Blocage d'un sous-ensemble de bloc

5.2.3.1 Assemblage fortement topologiquement autobloquant

Soit un assemblage composé de N blocs. Supposons que chacun de ses blocs soit bloqué en translation. Chaque sous-ensemble connexe composé de k blocs ($k < N$) n'est pas nécessairement bloqué en translation. C'est-à-dire qu'en considérant un tel sous-ensemble comme un unique solide rigide, il peut exister un mouvement de translation pour le retirer de l'assemblage. Cette remarque permet d'introduire cette nouvelle définition :

Définition 4 : Un assemblage de N blocs est **fortement** topologiquement autobloquant en translation si chaque sous-ensemble connexe de k blocs pour $k \in \{1, \dots, N\}$ est bloqué en translation (hors blocs frontières).

La Fig. 5.19 illustre ce propos. Le maillage est un maillage de carrés et l'inégalité 5.6 est respectée pour chacune de ses faces. Chaque bloc est donc bloqué en translation. En revanche, il existe une translation possible qui permet de retirer un sous-ensemble de quatre blocs sans être bloqué par les blocs voisins. Cette translation est représentée par la flèche noire dans la Fig. 5.19.

-45°	+30°	-30°	+45°
+45° +45°	-45° -45°	+45° +60°	-60° -45°
-45° +45°	+30° -30°	-30° +30°	+45° -45°
-30° -30°	+30° +60°	-60° +45°	-45° +60°
+45° -45°	-30° +30°	+60° -60°	-45° +45°
+30° +30°	-30° -30°	+30° +60°	-60° -30°
-45° +45°	+60° -60°	+45° -45°	+60° -60°
-45° -45°	+45° +45°	-45° -45°	+45° +45°
+45° +45°	-30° -30°	+60° +60°	-45° -45°

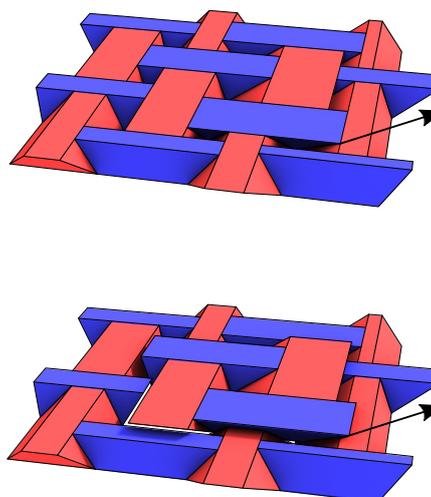
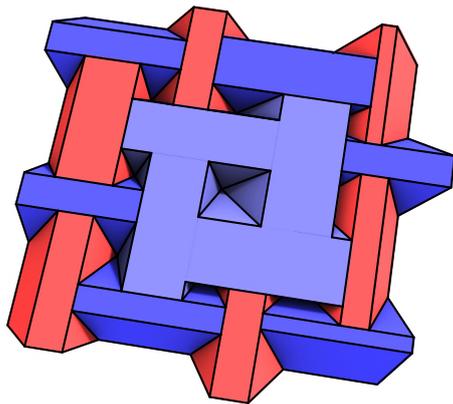


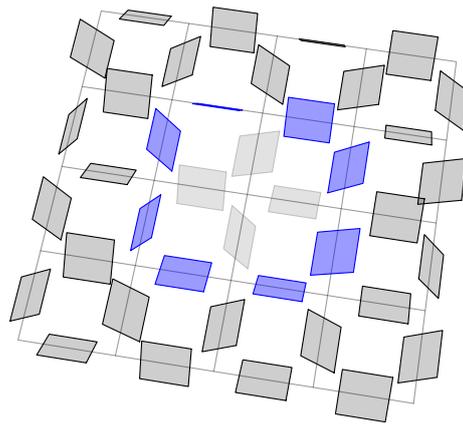
FIGURE 5.19 – Cet assemblage est topologiquement autobloquant (chaque bloc est bloqué en translation) mais n'est pas **fortement** topologiquement autobloquant. Il existe un sous-ensemble de quatre blocs non bloqué en translation. La flèche noire représente un déplacement de translation possible pour extraire ce sous-ensemble.

5.2.3.2 Conditions de blocage

Soit un sous-ensemble connexe S constitué de k blocs et \mathbb{P} l'ensemble des plans P_i de la frontière de ce sous-ensemble. Pour le sous-ensemble de quatre blocs de la Fig. 5.20a, l'ensemble \mathbb{P} – composé de huit plans – est représenté Fig. 5.20b. Si le sous-ensemble est composé d'un unique bloc ($k = 1$), alors $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$.



(a) Le sous-ensemble est considéré comme un solide unique (non convexe).



(b) Plans permettant de construire les blocs de l'assemblage. L'ensemble \mathbb{P} des plans de la frontière du sous-ensemble est représenté par les plans en bleu.

FIGURE 5.20

En adaptant la **Définition 1**, le sous-ensemble S , considéré comme un solide unique, est bloqué en translation s'il n'existe aucun mouvement infinitésimal de translation rectiligne possible permettant de l'enlever de l'assemblage en considérant ses voisins comme fixes.

Le sous-ensemble de la Fig. 5.20 touche ses voisins selon deux types de contact :

1. des contacts plan-plan (les plans plans appartenant à \mathbb{P}).
2. des contacts ponctuels arête-arête.

Un contact ponctuel arête-arête est par exemple représenté par le point Q dans la Fig. 5.21 : une arête du bloc V_1 est en contact ponctuel avec une arête du bloc V_3 . Dans un souci de clarté, l'assemblage représenté est le même que précédemment mais amputé de trois blocs. Considérons désormais le déplacement du sous-ensemble constitué de quatre blocs selon le vecteur \vec{u} également représenté sur la figure. Aucun plan de \mathbb{P} (ici de cardinal 6) n'empêche ce déplacement. En revanche, le contact ponctuel au point Q l'empêche.

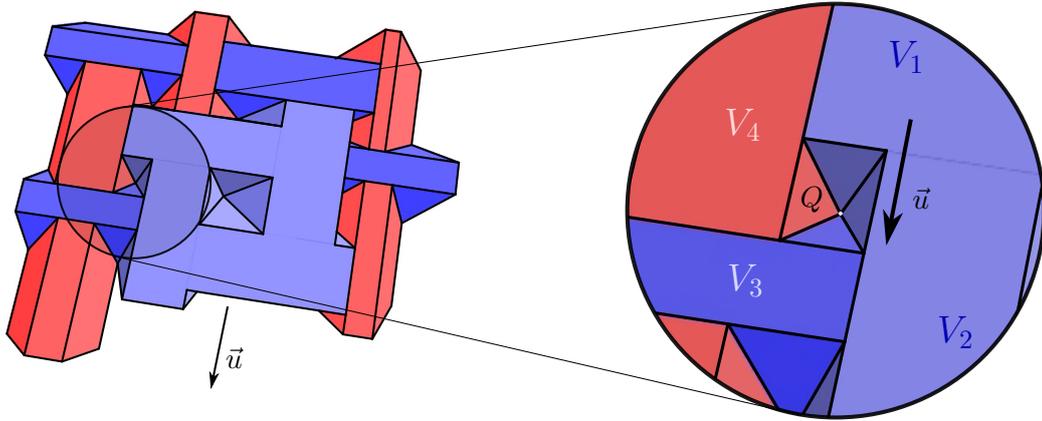


FIGURE 5.21 – Le déplacement selon le vecteur \vec{u} du sous-ensemble composé de quatre blocs n'est pas possible. Il est bloqué par le contact ponctuel arête-arête au niveau du point Q – contact entre une arête du bloc V_1 et une arête du bloc V_3 . En revanche, aucun contact plan-plan empêche ce déplacement.

Nous avons alors l'implication (et non l'équivalence) suivante :

$$\begin{array}{c}
 \forall \vec{u} \in E^*, \exists P \in \mathbb{P}, \text{ de normale } \vec{n}, \text{ tel que } \vec{u} \cdot \vec{n} < 0 \\
 \Downarrow \\
 \text{Le sous-ensemble connexe est bloqué en translation}
 \end{array}
 \tag{5.7}$$

Cette implication signifie que, pour tout vecteur, s'il existe un contact plan-plan qui empêche le déplacement selon ce vecteur, alors le sous-ensemble est bloqué. En revanche, comme nous venons de le voir, le fait que le sous-ensemble soit bloqué ne signifie pas nécessairement qu'un contact plan-plan empêche le déplacement, il pourrait s'agir d'un contact ponctuel arête-arête. Cette implication est une généralisation à un sous-ensemble de bloc de l'équivalence Eq. 5.1 relative à un unique bloc convexe.

L'équivalence 5.3 donne une condition sur les normales de quatre plans pour que ces derniers empêchent toutes les translations possibles. Les conditions ne portent que sur les normales aux plans. On peut donc les appliquer sur le sous-ensemble de blocs vu comme un solide unique : s'il existe quatre plans de \mathbb{P} qui vérifient le membre de droite de Eq. 5.3, alors le sous-ensemble est bloqué en translation. La réciproque n'est pas vraie ; il est envisageable que le sous-ensemble soit bloqué par un groupe de 5 plans ou plus (plans appartenant à \mathbb{P}) sans être bloqué par n'importe quel groupe de 4 plans de \mathbb{P} , ou bien que le blocage a lieu à cause d'un contact ponctuel arête-arête. Cette remarque nous donne donc un moyen de vérifier qu'un sous-ensemble est bloqué en translation. Pour qu'un assemblage soit topologiquement **fortement** autobloquant il faudrait réussir à vérifier le blocage pour tous les sous-ensembles connexes possibles de l'assemblage. Cela

n'a pas été étudié plus en amont dans cette thèse.

Remarque : Nous avons également une condition nécessaire – mais non suffisante – pour qu'un sous-ensemble S ne soit pas bloqué en translation : il faut que pour tout groupe de quatre plans de \mathbb{P} , l'inégalité de Eq. 5.3 ne soit pas respectée. Si \mathbb{P} est composé de 8 plans, il y a $\binom{8}{4} = 70$ groupes différents de 4 plans. Pour l'assemblage de la Fig. 5.19, parmi ces 70 groupes, aucun groupe de quatre plans ne respecte l'inégalité de Eq. 5.3.

5.2.4 Limites

Dans cette section, nous nous sommes intéressés aux conditions de blocage en translation d'un assemblage obtenu à partir d'un maillage de quadrangles plans. Néanmoins, nous avons uniquement étudié le cas des blocs à faces planes et nous n'avons pas considéré le blocage en rotation. Les assemblages *fortement* topologiquement autobloquants ont également été introduits. Cependant, puisque des blocages ponctuels arête-arête peuvent intervenir pour les sous-ensembles de blocs, le blocage (ou non-blocage) d'un sous-ensemble de blocs n'a été abordé que superficiellement.

5.3 Applications

5.3.1 Voûte à double courbure positive

5.3.1.1 Génération du maillage

Dans cette première application, la surface considérée est une surface à double courbure positive. Plus précisément, il s'agit d'une surface moulure (Fig. 5.22). Cette dernière est obtenue par le balayage d'une courbe plane nommée *génératrice* le long d'une autre courbe plane nommée *rail*. Ce procédé permet d'avoir un maillage de quadrangles plans. Dans cet exemple, le rail est une courbe de chaînette.

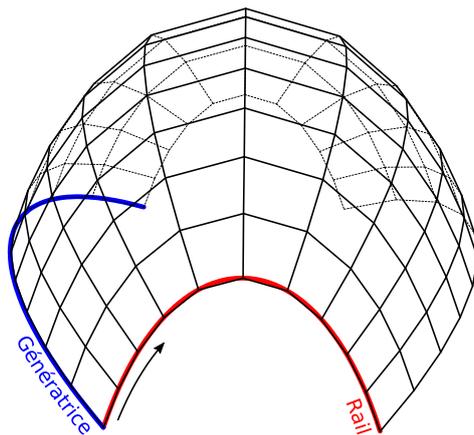


FIGURE 5.22 – Maillage de quadrangles plans d'une surface à double courbure positive

5.3.1.2 Génération d'un assemblage topologiquement autobloquant

Ce maillage permet de créer un assemblage topologiquement autobloquant en utilisant la généralisation – introduite dans ce chapitre – de la procédure *moving cross-section*; cette généralisation permettant de concevoir des assemblages topologiquement autobloquants à partir de maillages non réguliers de surfaces non planes. Le membre de droite de Eq. 5.3 est vérifié pour chaque bloc, prouvant de ce fait que l'assemblage est topologiquement autobloquant. Concernant la troncature de chaque bloc (cf. Sec. 5.1.2), les plans P_{h_1} et P_{h_2} ont été utilisés avec $h_1 = 0$ et $h_2 < 0$ (comme sur la Fig. 5.11b). L'assemblage obtenu est représenté Fig. 5.23.

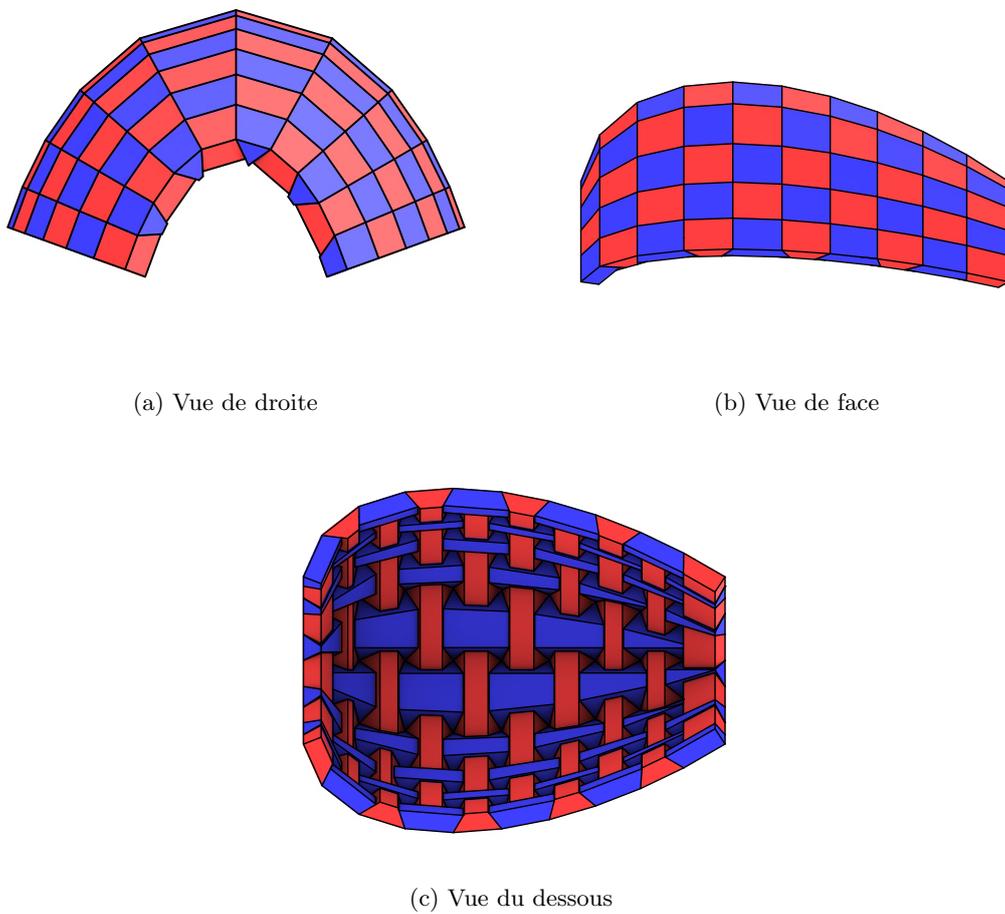


FIGURE 5.23 – Voûte à double courbure positive topologiquement autobloquante

5.3.1.3 Calcul de stabilité

Comme exposé au Chapitre 3, la méthode d’analyse de stabilité la plus appropriée est le calcul aux éléments discrets ou bien le calcul aux éléments finis avec des blocs distincts. Nous adoptons cette deuxième approche en effectuant l’analyse à l’aide du logiciel Abaqus grâce à un calcul explicite quasi-statique. Chaque bloc est modélisé séparément comme un solide déformable; les blocs pouvant rentrer en contact les uns avec les autres. À ce titre, une loi d’interaction régissant le contact des blocs entre eux est également utilisée.

Géométrie Pour l’analyse, la longueur de la voûte est de l’ordre de 10 m et sa largeur de l’ordre de 8 m (Fig. 5.24). Son épaisseur est de l’ordre de 0.44 m. L’espace de modélisation est un espace 3D et puisque la voûte n’est pas symétrique, elle est considérée dans son ensemble.

Matériau Le matériau utilisé pour la simulation est homogène et élastique linéaire isotrope. Les paramètres matériaux utilisés sont reportés dans la Table. 5.1 et sont proches de ceux d’une pierre calcaire dure ou d’un granite tendre (Serratrice and Durville, 1997; Chalhoub, 2010).

TABLE 5.1 – Propriétés du matériau utilisé pour la simulation Abaqus

Module d’Young	E	20000	MPa
Coefficient de Poisson	ν	0.27	
Masse volumique	ρ	2700	kg.m ⁻³
Résistance à la traction	R_t	5	MPa

Type d’analyse Le calcul est effectué grâce à une analyse de type explicite (*Abaqus/Explicit*). Ce type d’analyse consiste en une intégration en temps des équations de mouvement de chaque bloc par un schéma d’intégration par différences finies centrées. Ce type d’intégration est stable à condition que l’incrément de temps Δt soit inférieur à une certaine valeur Δt_{max} . Δt_{max} dépend de la taille du plus petit élément fini et de la vitesse des ondes acoustiques dans le matériau – vitesse dépendante de la densité de ce dernier. Ainsi, plus le maillage est raffiné, plus les éléments sont petits, plus Δt_{max} est petit et plus le nombre d’incrément nécessaires est grand, allongeant ainsi le temps de calcul. Puisque nous effectuons une analyse quasi-statique, nous ne nous intéressons pas aux effets dynamiques. Ainsi, nous pouvons augmenter artificiellement la masse volumique du matériau afin d’augmenter Δt_{max} . Ce procédé qui accélère les analyses s’appelle le *mass scaling* et est particulièrement adapté pour les analyses quasi-statiques dans lesquelles l’énergie cinétique est faible devant l’énergie interne. Nous avons utilisé ce procédé.

Conditions aux limites Les assemblages topologiquement autobloquants présentés aux Chapitre 2 sont contraints à leur périphérie. Ici, on fait l’hypothèse que les bords

libres de l'assemblage, représentés par les surfaces rouges et bleues sur la Fig. 5.24, sont encastrés; c'est-à-dire que les trois degrés de liberté des nœuds du maillage associés à ces surfaces sont fixés à zéro. L'encastrement est également représenté sur la Fig. 5.25. Il s'agit d'une hypothèse assez forte. La surface rouge est celle directement en contact avec le sol. Faire cette hypothèse peut revenir à considérer les fondations comme infiniment rigides. Les surfaces bleues en revanche ne sont pas au contact du sol. On peut envisager qu'elles sont en contact avec une armature métallique ou en béton armé, à l'image d'une maçonnerie chaînée.

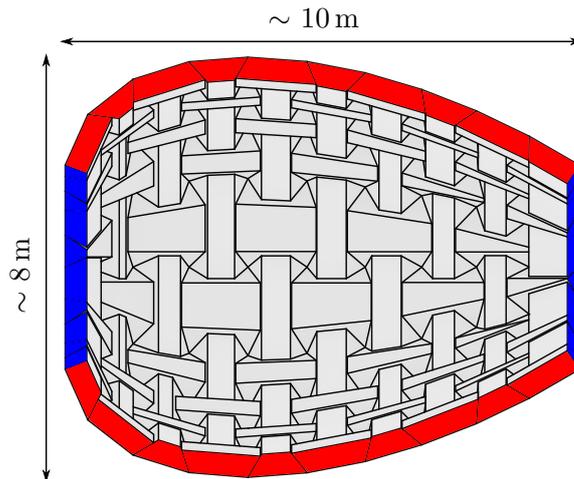


FIGURE 5.24 – Voûte vue de dessous. Dans l'analyse, les surfaces rouges et bleues sont encastrées. L'épaisseur des blocs est de 0.44 m.

Chargement Le cas de charge considéré est composé des deux charges suivantes :

- le poids propre
- un chargement dissymétrique : une pression de 50 kPa appliquée sur une surface d'environ 10 m^2 , représentée par les faces rouges sur la Fig. 5.25.

Le poids propre est d'abord appliqué graduellement sur une durée de 100 s afin d'éviter des effets de choc et d'avoir une convergence plus rapide avec moins d'oscillations. Au bout de 100 s le chargement vaut donc 100% du poids propre et ce chargement est maintenu jusqu'à la fin de l'analyse. Une fois que la voûte a trouvé son état d'équilibre sous poids propre, le chargement dissymétrique est également appliqué graduellement sur une durée de 300 s. Lors de cette deuxième partie de l'analyse, le chargement total consiste donc en un chargement de poids propre et un chargement dissymétrique.

Modélisation du contact Le frottement considéré est de type Coulomb non associé (Chap. 3). Les valeurs utilisées sont reportées Tab. 5.2.

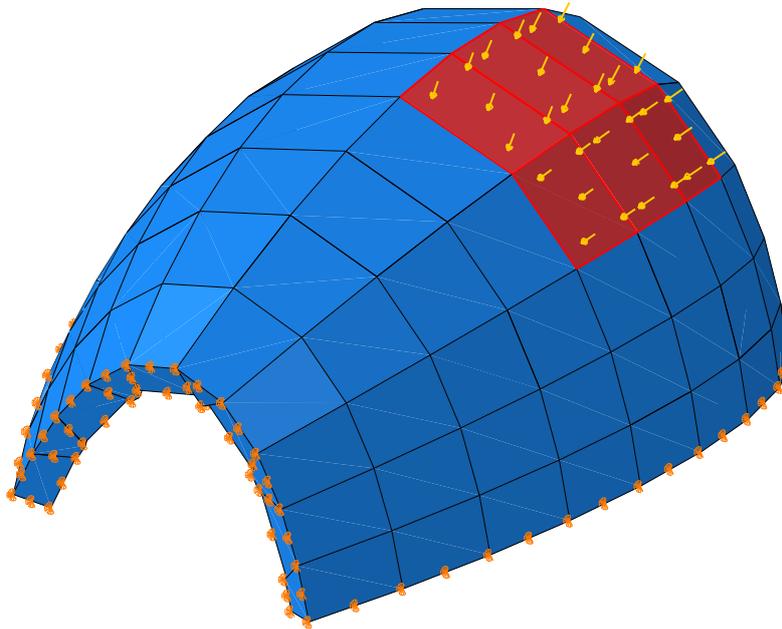


FIGURE 5.25 – Chargement dissymétrique : une pression de 50 kPa est appliquée sur les faces rouges.

TABLE 5.2 – Propriétés du contact de Coulomb

Cohésion	C	0
Angle de frottement	ϕ	30°
Angle de dilatance	ψ	0°

Dans Abaqus, l'interaction de contact entre les différents blocs que nous avons utilisée est du type *General Contact*. Le comportement normal est modélisé par une interaction de type *Hard Contact* qui utilise une méthode de pénalité. Cette méthode équivaut à introduire un ressort élastique entre les surfaces des blocs en contact. La rigidité de ce ressort est choisie automatiquement par Abaqus afin que l'interpénétration au niveau de l'interface de contact soit minimale tout en ayant un temps de calcul raisonnable. La séparation des surfaces en contact est possible dans cette modélisation. Une méthode de pénalité est également utilisée pour le comportement tangentiel.

Maillage et type d'éléments finis utilisés Dans un calcul aux éléments finis, le taux de distorsion des éléments est représentatif de la qualité du maillage. Plus ce taux est grand, moins la précision des résultats est bonne. À cause de la géométrie particulière des blocs (avec des angles aigus), mailler avec des éléments hexaédriques conduit à beaucoup d'éléments distordus. Nous avons alors maillé les blocs avec des éléments tétraédriques.

De plus, dans des analyses impliquant du contact, utiliser des éléments quadratiques peut poser problème car les forces nodales pour un même éléments sont différentes. Il est alors conseillé d'utiliser des éléments linéaires. Les éléments que nous avons utilisés sont donc des éléments C3D4 tétraédriques linéaires. Afin de faire une analyse de convergence, différentes densités de maillages ont été utilisées. Un exemple de bloc est représenté Fig. 5.26 pour le maillage le plus dense utilisé. Il est composé de 19238 éléments et 4037 nœuds.

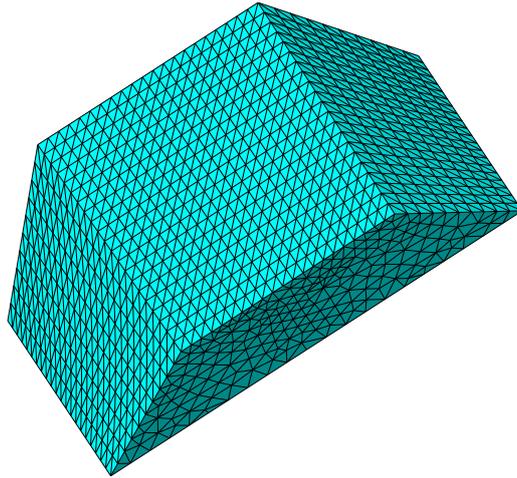


FIGURE 5.26 – Exemple de maillage d'un bloc avec un maillage dense (19238 éléments et 4037 nœuds).

5.3.1.4 Résultats

Validité de l'analyse quasi-statique Afin de vérifier que l'analyse est bien quasi-statique, il faut s'assurer que les effets dynamiques sont négligeables. Au cours de l'analyse, l'énergie cinétique est toujours inférieure à 4% de l'énergie interne de déformation. De plus, suite à l'application du poids propre, le système trouve un état d'équilibre : l'énergie de déformation et l'énergie cinétique sont stables et cette dernière vaut moins de 1% de l'énergie de déformation (Fig. 5.27). De même, après l'application graduelle du chargement dissymétrique (entre 300 s et 600 s), l'énergie de déformation se stabilise et, en fin de simulation, l'énergie cinétique vaut moins de 1% de cette dernière. Ces considérations nous permettent de valider l'hypothèse d'une simulation quasi-statique et de justifier l'utilisation du *mass scaling*.

Analyse de convergence Différentes densités de maillage ont été utilisées afin de faire une étude de convergence. Deux critères ont été retenus : un critère global, l'énergie interne de déformation et un critère plus local, la flèche maximale. Les résultats de cette analyse de convergence sont représentés Fig. 5.28. La convergence n'est pas totalement

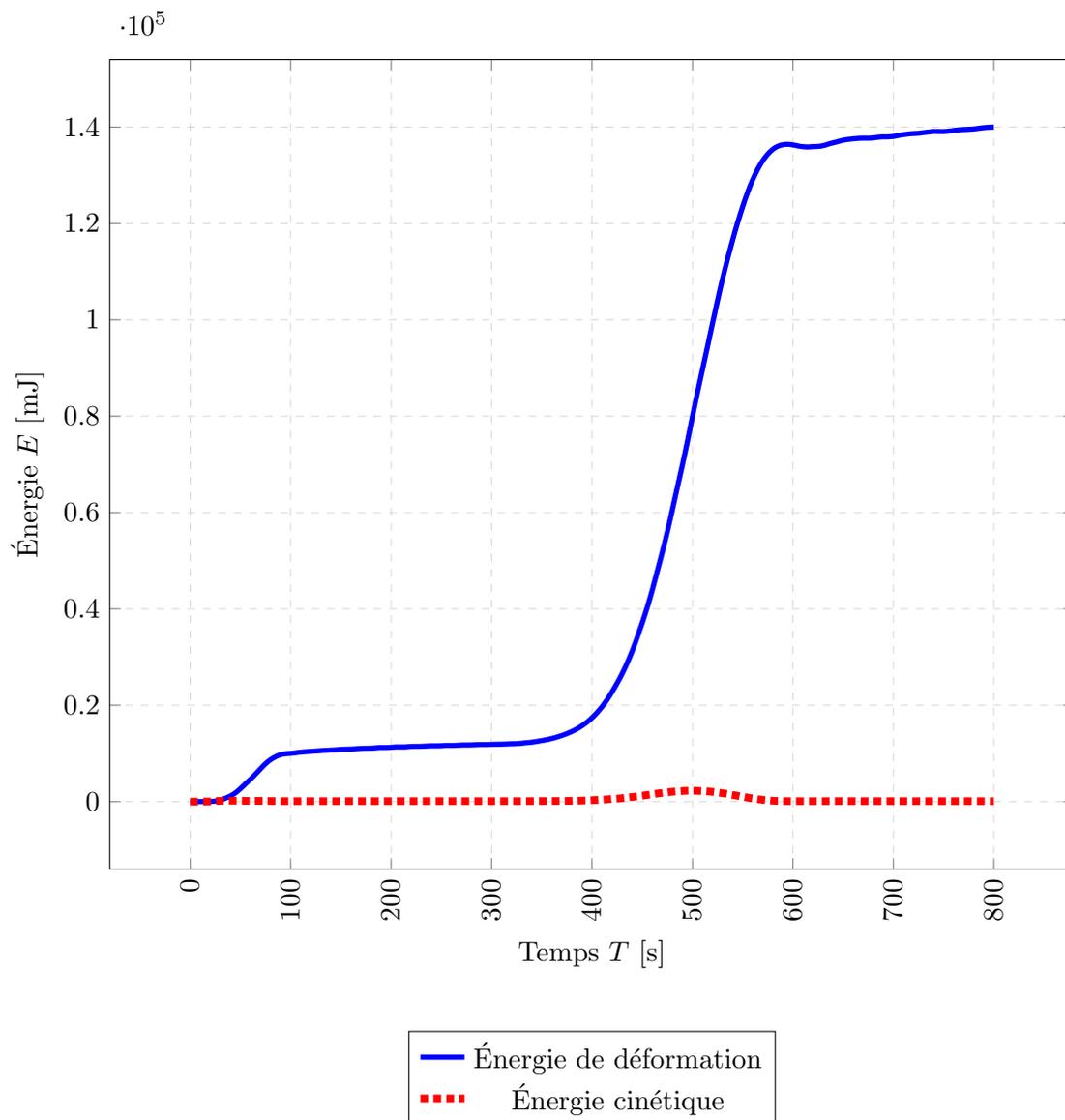


FIGURE 5.27 – Énergie de déformation et énergie cinétique lors du chargement de la voûte. La voûte est chargée graduellement par son poids propre entre 0s et 100s. Une fois l'état d'équilibre trouvé, le chargement dissymétrique est appliqué graduellement entre 300s et 600s.

atteinte mais est raisonnable. De plus, utiliser un maillage encore plus fin mène à des temps de calcul long. Nous n'avons donc pas raffiné plus le maillage.

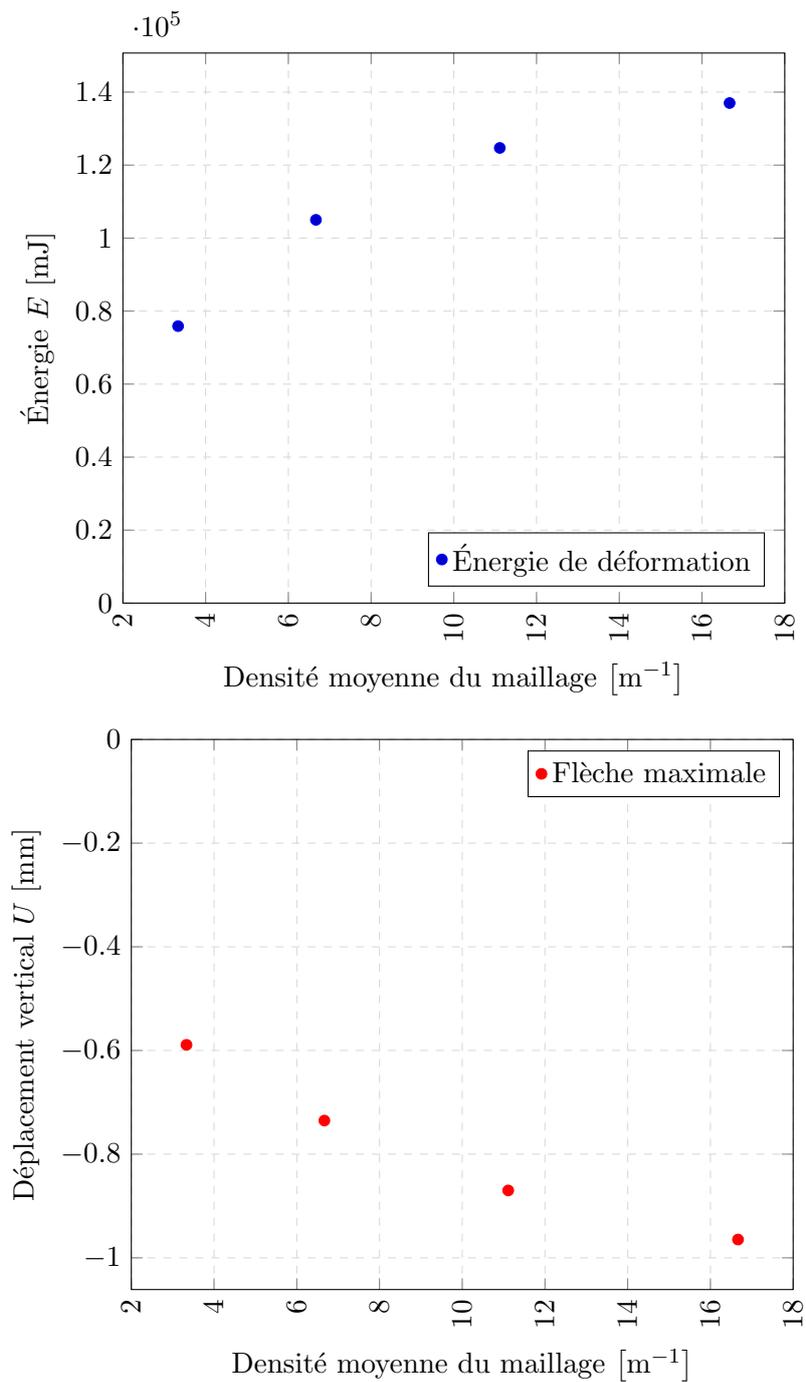


FIGURE 5.28 – Analyse de convergence

Déplacement et contraintes Les résultats qui nous intéressent sont la flèche maximale et les contraintes maximales. Les déplacements verticaux sont représentés Fig. 5.29a

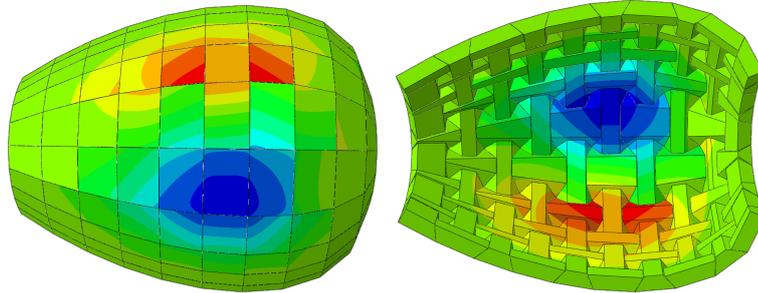
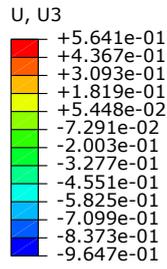
pour le maillage fin. Il est d'usage de représenter les contraintes sous la forme de contraintes de Von Mises mais le matériau utilisé ici s'apparente à de la pierre qui est fragile principalement en traction. En plus des contraintes de Von Mises (Fig. 5.29b), nous avons alors étudié les contraintes principales maximales et minimales (Fig. 5.29c et 5.29d). Les contraintes maximales atteintes se trouvant sur les surfaces extérieures des blocs, nous avons uniquement représenté les champs de contraintes sur les surfaces extérieures de la voûte.

Concernant le déplacement vertical, la flèche maximale est un déplacement vers le bas de 0.96 mm au niveau de la zone d'application du chargement dissymétrique. La partie opposée de la voûte en revanche se soulève légèrement de 0.56 mm. Concernant les efforts, des concentrations de contraintes ont lieu au niveau des coins des blocs, là où quatre blocs se rejoignent. La concentration de contrainte la plus importante est visible Fig. 5.31, la zone représentée correspondant au rectangle blanc de la Fig. 5.31. Plus le maillage est fin, plus la contrainte maximale est élevée mais la zone concernée est également plus localisée. À part ces points singuliers de concentration de contrainte, la contrainte maximale dans le reste de la structure – aussi bien celle de Von Mises que celle principale maximale – ne dépasse pas la valeur de résistance en traction du matériau considéré (5 MPa). Nous pouvons considérer que la structure ainsi modélisée supporte le chargement imposé.

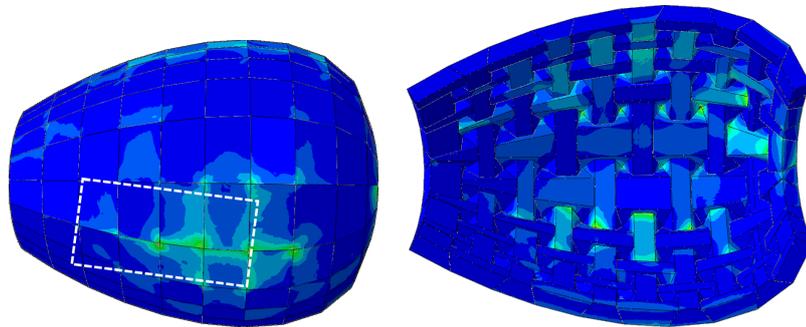
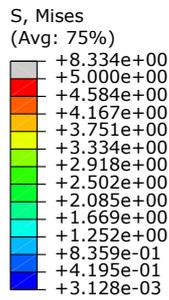
Discussion La rigidité de la structure ainsi modélisée est conséquente (une flèche de 1 mm pour une *portée* de 8 m). Afin de vérifier la validité de notre modélisation, nous avons également étudié la structure continue équivalente, en utilisant une modélisation de coque. Les résultats sont présentés Fig. 5.30 et semblent cohérents avec ceux trouvés dans le cas discret. La flèche obtenue est de l'ordre de 0.1 mm, soit 10 fois moins que pour le modèle discret. Les champs de contraintes ont une apparence relativement similaire à ceux de la Fig. 5.29 (l'échelle de couleur n'est pas la même, notamment à cause des concentrations de contraintes dans le modèle discret).

Nous expliquons la très grande rigidité de la structure par plusieurs arguments. Premièrement, par l'absence de jeu. En effet, cette modélisation considère des géométries parfaites. Les faces sont parfaitement planes et en contact les unes avec les autres. Les écarts dimensionnels entre la géométrie réelle et la géométrie idéale peuvent cependant être grands et réduire la rigidité de la structure. La deuxième raison avancée pour expliquer la grande rigidité est l'élanement relativement faible de la structure et les conditions aux limites fortes (encastrement). Enfin, la modélisation des contacts par une interaction de type *Hard Contact* confère une rigidité supérieure à celle que l'on aurait avec une modélisation de type *soft contact* ou en modélisant le mortier.

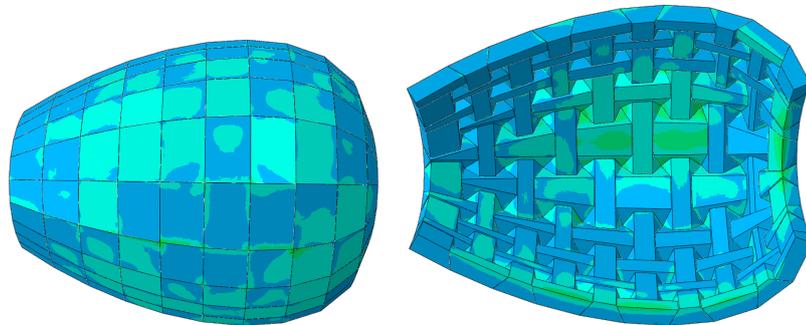
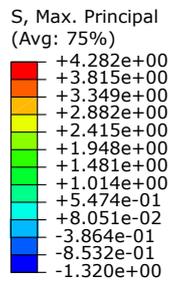
La modélisation pourrait être améliorée de diverses manières. Premièrement, en utilisant des conditions aux limites moins fortes; par exemple, en modélisant des appuis avec une rigidité donnée. Deuxièmement, en rajoutant des défauts dimensionnels. Enfin, en considérant une rigidité finie dans la modélisation du contact.



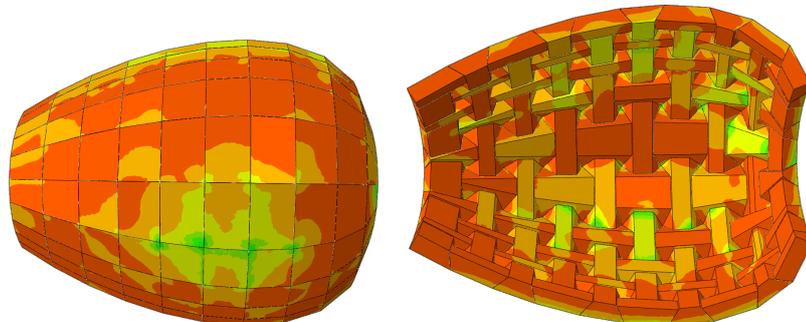
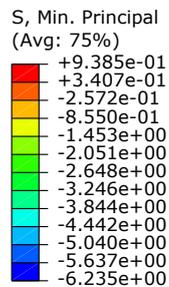
(a) Déplacement vertical (mm)



(b) Contraintes équivalentes de Von Mises (MPa). La zone située dans le rectangle blanc est représentée Fig. 5.31

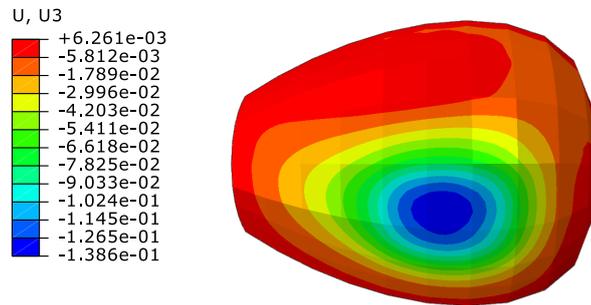


(c) Contraintes principales maximales (MPa)

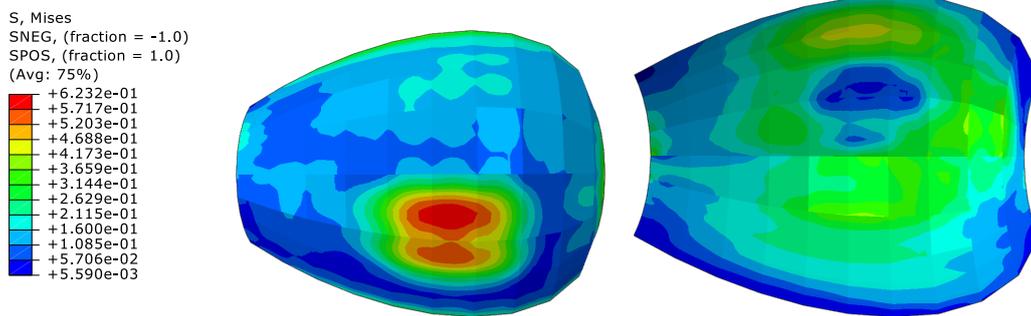


(d) Contraintes principales minimales (MPa)

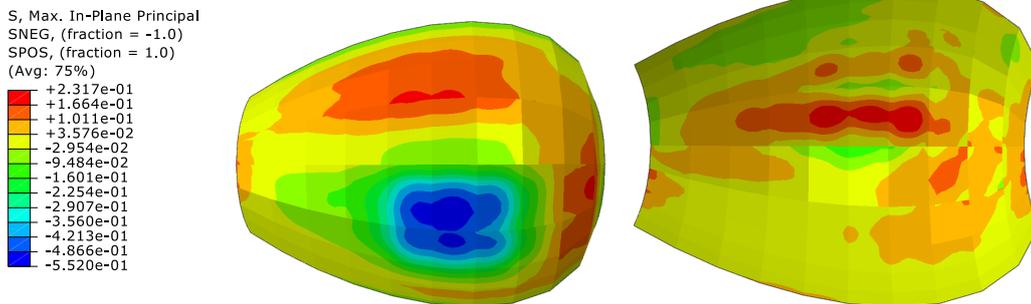
FIGURE 5.29 – Résultats pour le cas de charge : poids propre + chargement dissymétrique. Un déplacement positif correspond à un déplacement vers le haut. Une contrainte positive est une contrainte en traction.



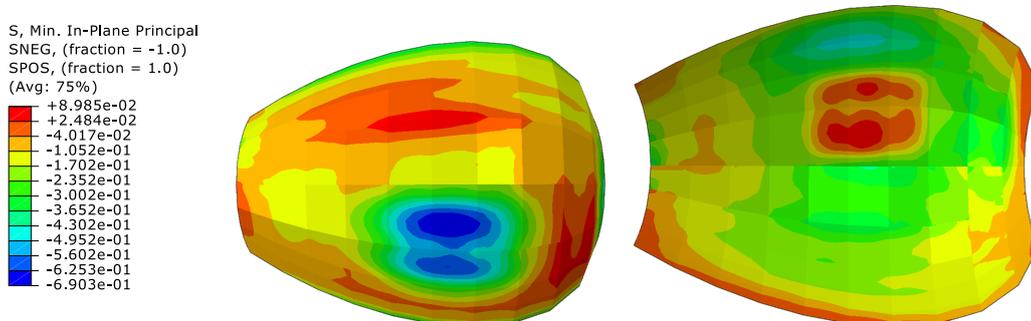
(a) Déplacement vertical (mm)



(b) Contraintes équivalentes de Von Mises (MPa).

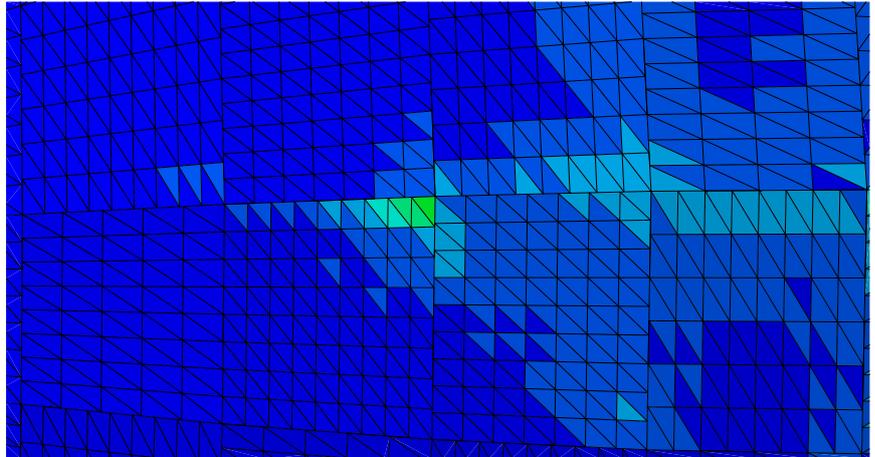


(c) Contraintes principales maximales (MPa)

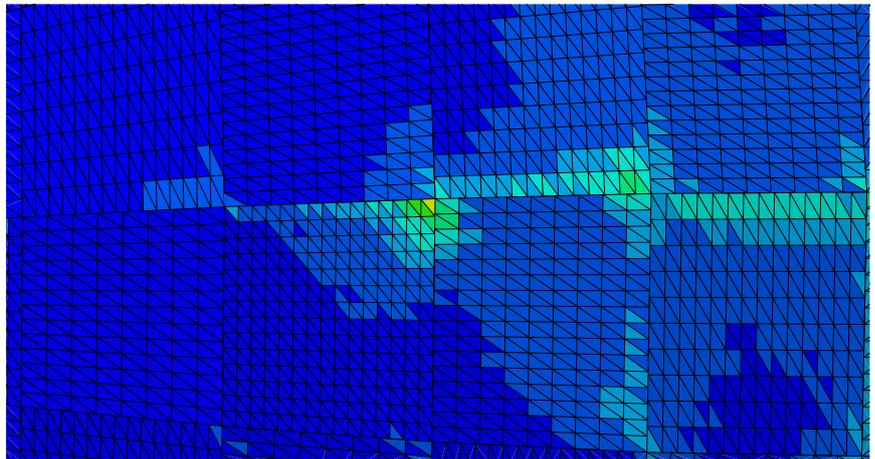
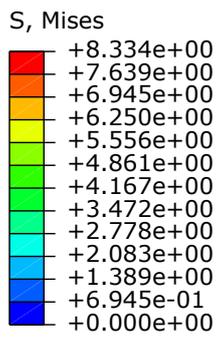


(d) Contraintes principales minimales (MPa)

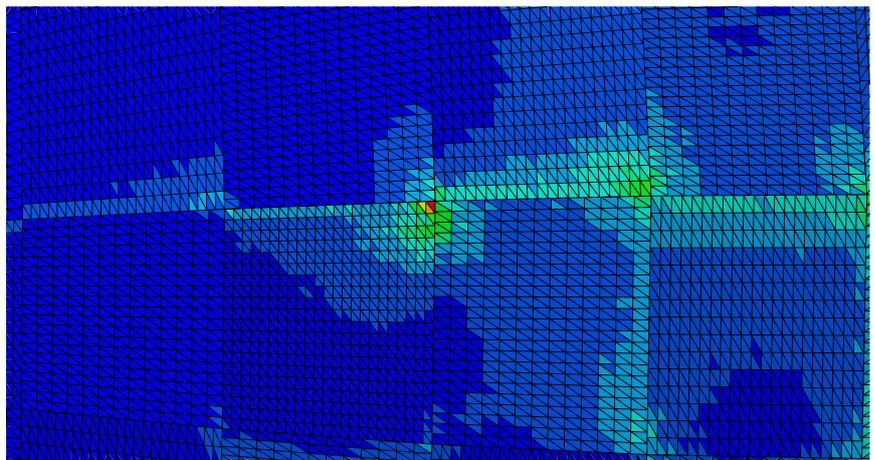
FIGURE 5.30 – Résultats pour une modélisation continue équivalente (modélisation de coque). Un déplacement positif correspond à un déplacement vers le haut. Une contrainte positive est une contrainte en traction.



(a) Maillage grossier (densité = 6.67 m^{-1})



(b) Maillage moyen (densité = 11.11 m^{-1})



(c) Maillage fin (densité = 16.67 m^{-1})

FIGURE 5.31 – Concentration de contrainte de Von Mises au coin du bloc. Unité : MPa

Prototype La stabilité d'une structure maçonnée est principalement un problème d'équilibre et de géométrie (Sec. 3.1). De ce fait, l'analyse de stabilité peut en première approche être effectuée à partir d'un modèle réduit. Nous avons donc imprimé en 3D un modèle réduit de la voûte de dimensions 20 cm×25 cm. Le matériau utilisé est de l'ABS, dont l'angle de frottement a été estimé à 25°. L'imprimante utilisée est une Stratasys Dimension Elite permettant d'imprimer avec une épaisseur de couche de l'ordre de 0.17 mm. Les blocs de la frontière ont été imprimés d'un seul tenant. Plus précisément, la frontière a été imprimée en quatre parties, chacune de ces parties consistant en plusieurs blocs *collés* ensemble (Fig. 5.32). Les deux premières parties de cette frontière sont constituées des blocs correspondants aux surfaces bleues de la Fig. 5.24 et les deux autres parties correspondent aux surfaces rouges. Afin de pouvoir encastrer ce modèle réduit dans une plaque de MDF de 6 mm d'épaisseur, le bas de ces blocs a été légèrement modifié. Ce modèle réduit est représenté Fig. 5.33. Il est stable sous poids propre et également lorsqu'on le charge légèrement.



FIGURE 5.32 – À gauche : sous-ensembles de blocs formant la frontière de la voûte. À droite : base en MDF.



FIGURE 5.33 – Modèle réduit

5.3.2 Corolle

Dans cette deuxième application, la surface considérée est à double courbure négative, il s'agit encore d'une surface moulure (Fig. 5.34). Le principe étudié dans cette deuxième application est la construction au dessus du vide sans coffrage. Pour cela, nous nous sommes inspirés de la construction sans coffrage des voûtes Byzantines (cf. Sec. 1.2.1) en utilisant les propriétés de blocage et de rigidité en flexion propres aux assemblages topologiquement autobloquants.

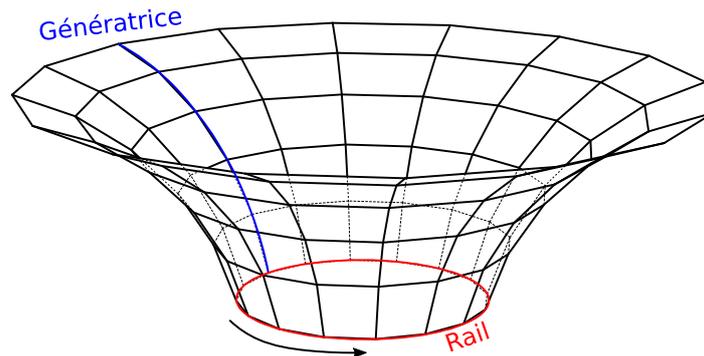


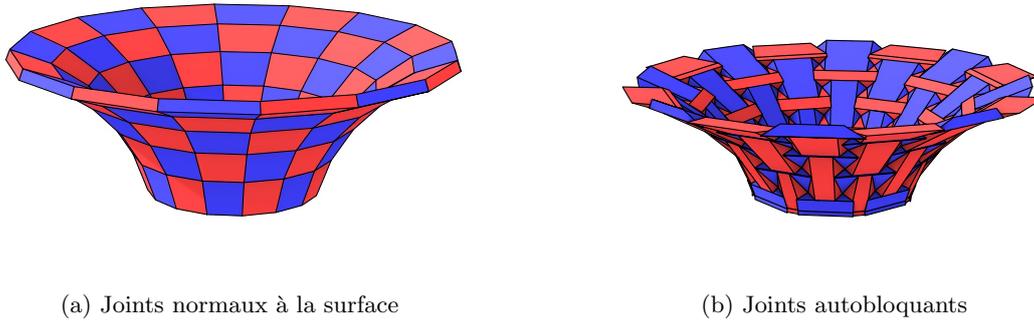
FIGURE 5.34 – Maillage de quadrangles plans

En effet, à partir de ce maillage de quadrangles plans, il est possible de modéliser des blocs de différentes manières :

1. une manière conventionnelle avec une découpe de joints normaux à la surface – suivant le principe de normalité exposé en Sec. 2.1. La structure qui en découle est représentée Fig. 5.35a. Sans la présence de mortier ou de clés de blocage, cette structure n'est clairement pas stable sous poids propre. En effet, les joints ne peuvent pas reprendre des efforts de traction et pourtant de nombreux cercles de traction sont présents (voir à ce propos la Fig. 5.45 de la section Sec. 5.3.2.2).
2. en inclinant les joints afin d'obtenir un assemblage topologiquement autobloquant. Une des structures possibles est représentée 5.35b. Il n'est pas évident de se prononcer sur la stabilité sous poids propre d'une structure ainsi obtenue. Cependant le blocage et la rigidité en flexion propres aux structures topologiquement autobloquantes permettent d'envisager la stabilité pour des inclinaisons de joints bien choisis.

Nous avons retenu cette deuxième manière de modéliser des blocs. La structure ainsi obtenue est composée de 80 blocs. Au lieu de donner des valeurs d'inclinaison de joints différentes pour chaque bloc et avoir 80 blocs tous différents, seulement 10 géométries de blocs différentes ont été utilisées en se basant sur l'axisymétrie du maillage (Fig. 5.36).

Nous nous sommes également inspirés de la construction des voûtes Byzantines pour pouvoir construire sans coffrage. Comme évoqué en Sec. 1.2.1, les Byzantins construi-



(a) Joints normaux à la surface

(b) Joints autobloquants

FIGURE 5.35 – Deux coupes de joints différentes pour une même surface

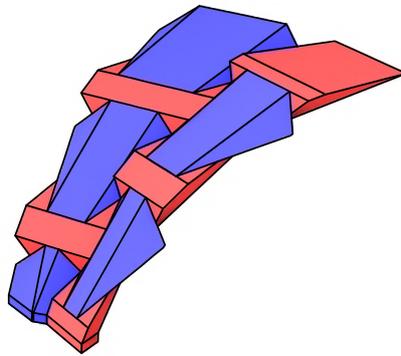


FIGURE 5.36 – La structure est composée des dix blocs représentés ici. Ce sous-ensemble de dix blocs est répété huit fois pour obtenir la structure entière de la Fig. 5.35b.

saient des voûtes comme une succession d'arcs accolés les uns aux autres. Une fois un arc construit, il était considéré comme stable. Il était ainsi possible de construire l'arc suivant en se reposant sur l'arc précédemment construit. Lors de la phase de construction de cet arc, chaque brique devait pouvoir supporter son poids propre. Pour cela, du mortier était utilisé, l'adhérence qu'il procurait permettant de porter le poids d'une brique. De plus, pour augmenter cette adhérence, les briques étaient légèrement inclinées. Notre méthode s'inspire de celle des Byzantins. La structure de la Fig. 5.35b peut être vue comme un empilement de cinq cercles concentriques de rayons croissants, chaque cercle étant constitué de 16 blocs (Fig. 5.37). L'idée est alors de construire un cercle en s'appuyant sur le cercle précédemment construit, en considérant qu'une fois le cercle construit en entier, il est suffisamment stable pour pouvoir supporter des chargements.

La méthode que nous avons utilisée pour pouvoir construire sans coffrage sur le cercle précédemment assemblé est illustrée Fig. 5.38. Il s'agit de concevoir les blocs de

telle sorte qu'un bloc sur deux soit supporté par les blocs du dessous (Fig. 5.38b). Cela revient à ce que le centre de gravité du bloc soit au dessus du bloc inférieur et non au dessus de vide, à la manière d'une construction en encorbellement (Fig. 5.39). Ces blocs pourraient néanmoins glisser, mais le glissement est empêché par les blocs inférieurs (Fig. 5.40). Une fois la moitié des blocs du cercle mis en place, les blocs manquants peuvent être construits en s'appuyant sur leurs voisins de gauche et de droite et non plus sur leur voisin du dessous (Fig. 5.38c).

Dans cette structure, la géométrie des blocs comporte des parties très fines avec des angles très aigus. Une telle géométrie n'est pas très adaptée pour une construction en pierre. Contrairement à l'application précédente, nous ne considérons plus une structure de 8 m sur 10 m mais une structure de taille réduite. Nous verrons un peu plus bas un prototype de cette structure dont le diamètre vaut environ 0.3 m à la base et 0.9 m au point le plus haut de la structure.

5.3.2.1 Analyse

La stabilité de ce prototype a été analysé sur Abaqus. Comme pour la structure précédente, l'analyse effectuée est explicite quasi-statique avec *mass scaling*.

Matériau Pour ce deuxième prototype, nous avons fait imprimer les blocs en 3D par l'entreprise VoxelJet. Le matériau utilisé est un sable GS14 dont la taille de grains moyenne est $14 \mu\text{m}$ et l'épaisseur de couche est de $300 \mu\text{m}$. Lors de l'impression, les grains sont collés entre eux à l'aide d'un liant. Les blocs ainsi obtenus ont une résistance relativement faible et sont alors durcis en les enduisant d'une résine époxy. Duballet (2014) a effectué des test de flexion sur des éprouvettes de ce matériau donnant une résistance à la traction de 27 MPa. L'angle de frottement a été estimé à 30° . Le module d'Young utilisé est celui d'un sable et la masse volumique a directement été calculée sur les blocs du prototype. Le matériau utilisé pour la simulation est considéré homogène et élastique linéaire isotrope. Les paramètres utilisés sont reportés dans la Table. 5.3.

TABLE 5.3 – Propriétés du matériau sable + liant + résine époxy

Module d'Young	E	8000	MPa
Coefficient de Poisson	ν	0.3	
Masse volumique	ρ	1500	kg.m^{-3}
Résistance à la traction	R_t	27	MPa

Paramètres modélisation La structure est encastrée à ses appuis et seul un chargement de poids propre est considéré. Ce dernier est appliqué graduellement entre 0s et 100s. Grâce à la symétrie de la structure et du chargement considéré, nous avons pu modéliser seulement un seizième de la structure afin d'accélérer les calculs (Fig. 5.41).

La modélisation du contact est similaire à l'analyse précédente et les éléments finis utilisés sont encore des éléments tétraédriques C3D4. La densité du maillage utilisé est

400 éléments par mètre. Un exemple de bloc maillé est représenté Fig. 5.42.

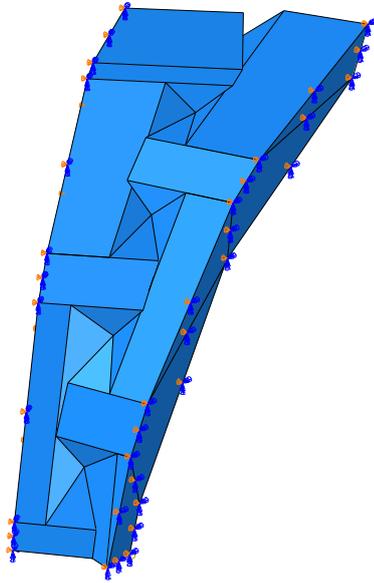


FIGURE 5.41 – Conditions aux limites. Encastrement en bas et conditions de symétries sur les bords.

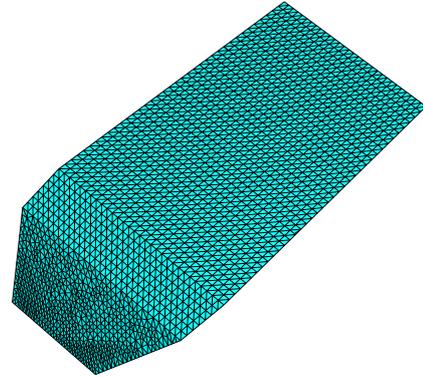


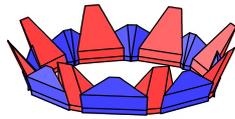
FIGURE 5.42 – Exemple de maillage dense (58428 éléments et 11879 nœuds).

5.3.2.2 Résultats

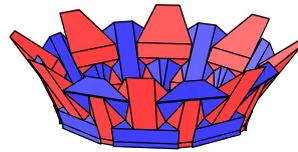
Validité de l'analyse quasi-statique Au cours de l'analyse, l'énergie cinétique vaut moins de 5% de l'énergie interne de déformation. Une fois l'énergie de déformation stable, l'énergie cinétique vaut moins de 0.1% de cette dernière. L'hypothèse d'une simulation quasi-statique est valide et l'utilisation du *mass scaling* est justifiée.

Analyse de convergence Les résultats de l'analyse de convergence sont représentés Fig. 5.43. La convergence est raisonnable.

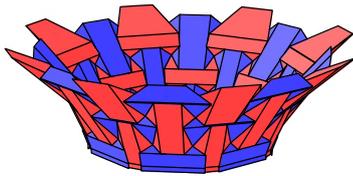
Déplacement et contraintes Les déplacements verticaux sont représentés Fig. 5.44a pour le maillage fin. La flèche maximale est de 1.1 mm, la hauteur de la structure étant de 248 mm. Les distributions des contraintes sont représentées Fig. 5.44. Des concentrations de contraintes ont lieu aux coins de certains blocs mais ne dépassent pas 2.8 MPa pour la contrainte équivalente de Von Mises ou 2.0 MPa pour la contrainte principale maximale, valeurs bien en dessous de la résistance en traction du matériau utilisé.



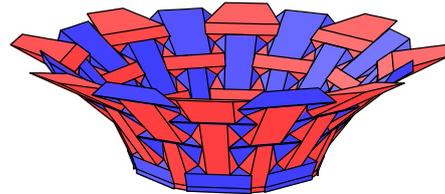
(a) Premier cercle



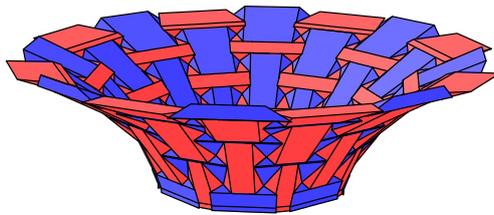
(b) Deuxième cercle



(c) Troisième cercle

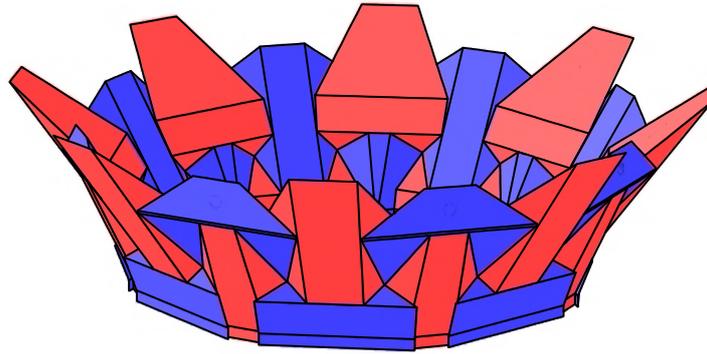


(d) Quatrième cercle

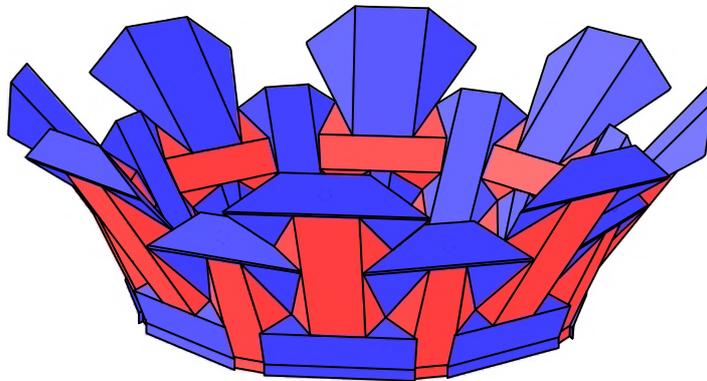


(e) Cinquième cercle

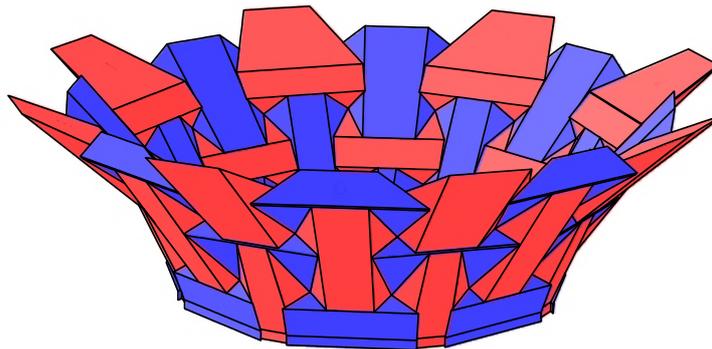
FIGURE 5.37 – La structure est composée de cinq cercles comprenant chacun 16 blocs.



(a) Les deux premiers cercles sont construits en entier



(b) Un bloc sur deux du troisième cercle est construit. Le centre de gravité de ces blocs est au dessus du voisins du dessous (cf. Fig. 5.39)



(c) Les blocs manquants sont posés afin de finir de construire le troisième cercle. Ils sont supportés par leurs voisins de gauche et de droite

FIGURE 5.38 – Construction d'un cercle composé de 16 blocs sans utiliser de coffrage

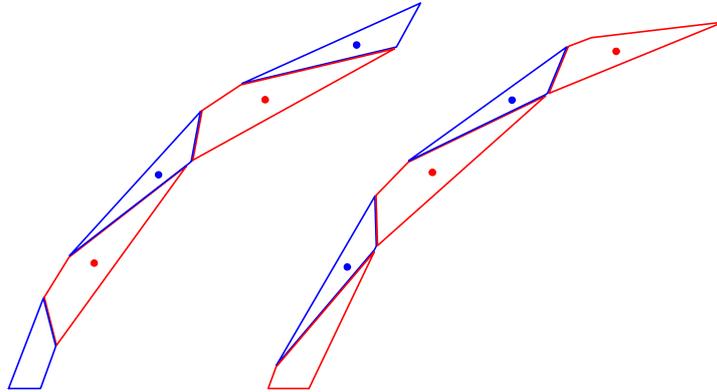


FIGURE 5.39 – Coupes des dix blocs de la Fig. 5.36. Les centres de gravité des blocs en bleu sont au dessus des blocs en rouge. Alors, en considérant les blocs rouges comme stables et en négligeant dans un premier temps le glissement, les blocs bleus sont stables sous l'effet de la gravité. Les centres de gravité des blocs rouges sont quant à eux au dessus du vide et ces blocs ne peuvent donc pas être maintenus par les blocs du dessous. Ils sont en fait supportés par leurs voisins de gauche et de droite.

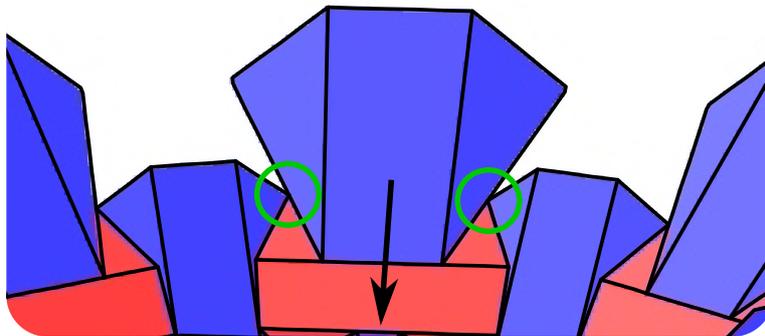


FIGURE 5.40 – Le bloc bleu est supporté par le bloc rouge du dessous mais, si le frottement n'est pas assez fort, il peut néanmoins glisser dans la direction de la flèche noire. Ce glissement est empêché par les blocs bleus du dessous au niveau des contacts arête-arête entourés en vert.

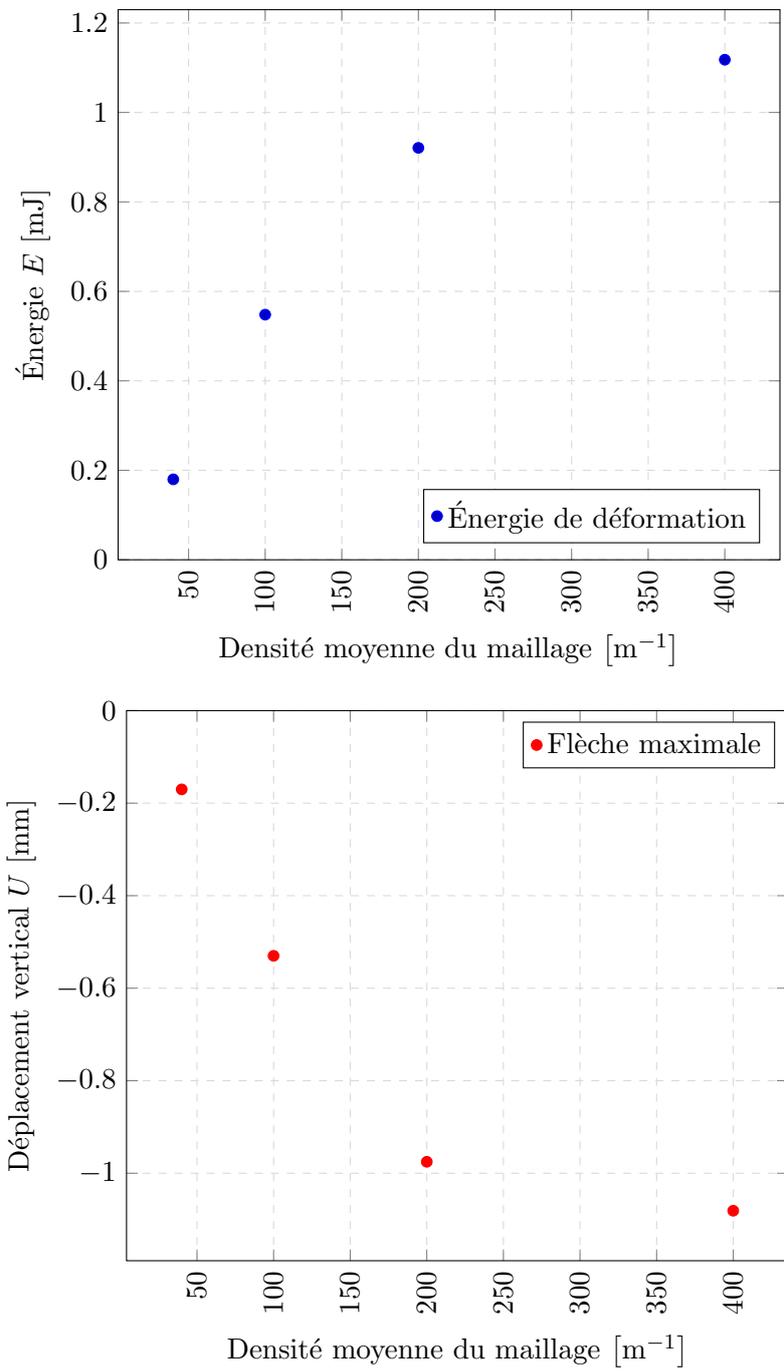
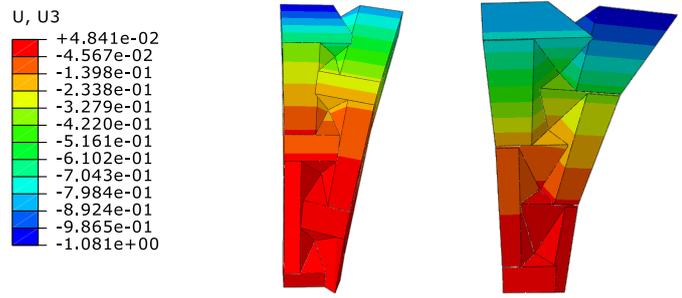
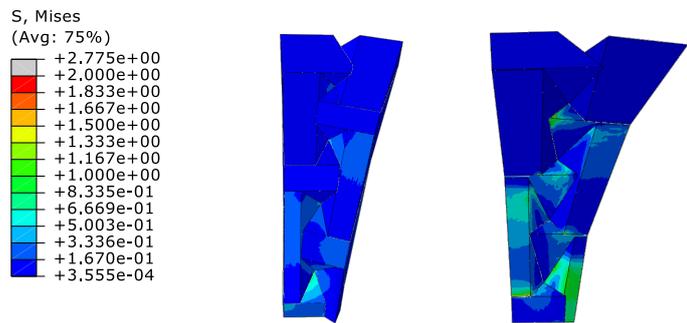


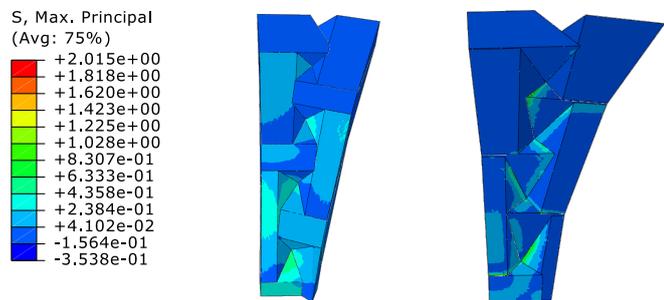
FIGURE 5.43 – Analyse de convergence



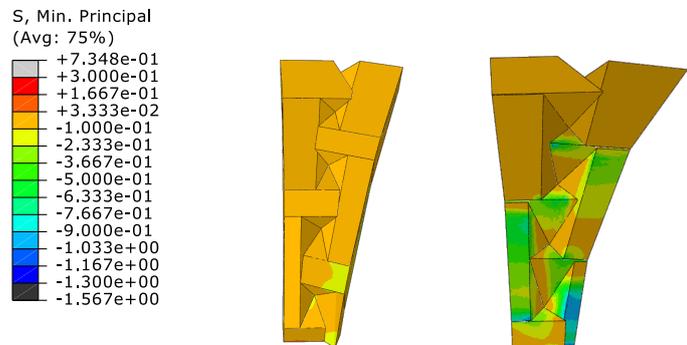
(a) Déplacement vertical (mm)



(b) Contraintes équivalentes de Von Mises (MPa).



(c) Contraintes principales maximales (MPa)



(d) Contraintes principales minimales (MPa)

FIGURE 5.44 – Résultats pour un chargement de poids propre. La structure est vue de face (milieu) et de derrière (droite). Un déplacement positif correspond à un déplacement vers le haut. Une contrainte positive est une contrainte en traction.

Discussion Une modélisation continue équivalente (modélisation de coque) a également été effectuée avec la structure encastree aux appuis. La figure Fig. 5.45 représente les efforts membranaires positifs dans la direction des parallèles. La partie colorée représente donc les cercles de traction. Ceux-ci apparaissent au niveau de la troisième couche de blocs.

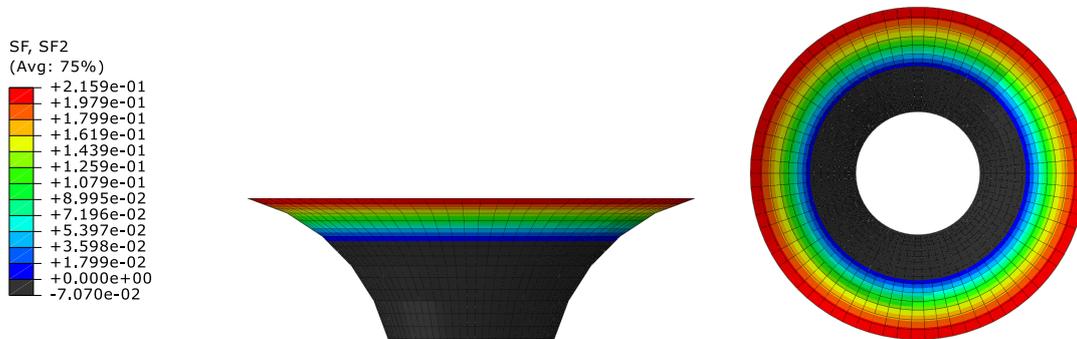


FIGURE 5.45 – Efforts membranaires de traction dans la direction des parallèles. La zone colorée correspond aux efforts de traction ($\text{N}\cdot\text{mm}^{-1}$).

Les blocs A, B, C et D de la figure Fig. 5.46 sont donc les blocs au niveau de ces cercles de traction. Afin d'essayer de comprendre le processus qui permet à ces blocs d'être en équilibre, les forces de contact entre les blocs A et B (Fig. 5.47) et entre les blocs C et D (Fig. 5.48) sont représentées. Il ressort de ces simulations que le contact est très localisé, soit dans un coin, soit sur une arête. Ce contact localisé dans un coin du bloc A est notamment visible sur le prototype que nous présentons au prochain paragraphe, comme illustré sur la Fig. 5.49.

La modélisation numérique par éléments finis que nous utilisons ne permet pas facilement de comprendre le comportement global de la structure. Notamment, il n'est pas évident d'expliquer comment sont repris les cercles de traction. Néanmoins, au vu des résultats de la modélisation continue équivalente, l'effort de traction maximal est de l'ordre de $0.2 \text{ N}\cdot\text{mm}^{-1}$ pour épaisseur de 20 mm, soit 4 N. Cet effort n'est pas énorme et nous supposons qu'il est repris par le frottement aux interfaces. L'influence du blocage / coincement et de l'inclinaison des interfaces sur l'équilibre ne ressort également pas dans cette analyse. Enfin, il est également possible qu'une comparaison avec une modélisation de coque ne soit pas la plus pertinente et que l'influence de cercles de traction soit minime. La structure se comporterait alors plus comme une succession de poutres en porte-à-faux, encastrees aux appuis. Une étude plus approfondie est nécessaire pour comprendre le comportement de cette structure et fait partie des perspectives possibles pour la suite de nos travaux.

Prototype Un prototype a été imprimé dans le matériau préalablement mentionné. Le diamètre vaut environ 30 cm à la base et 90 cm au point le plus haut de la structure. La dimension des blocs varie entre 10 cm et 20 cm, pour une épaisseur d'environ 2 cm.

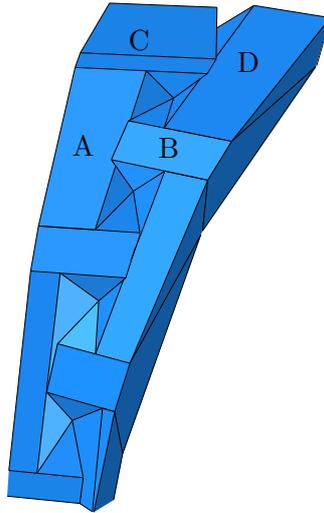
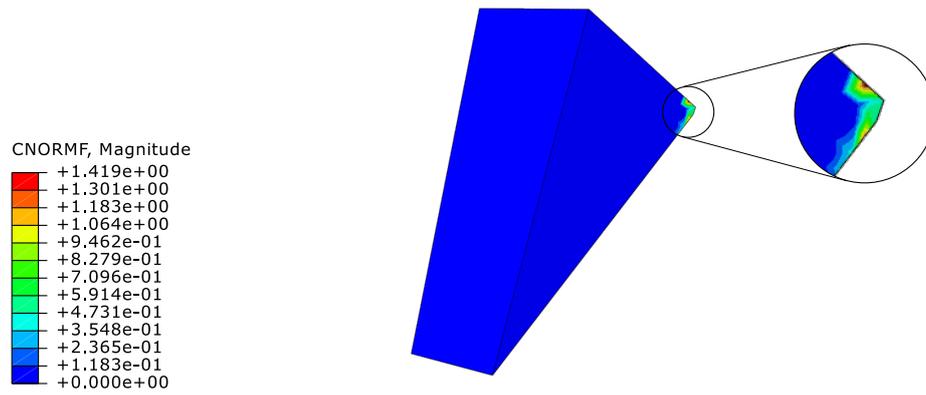
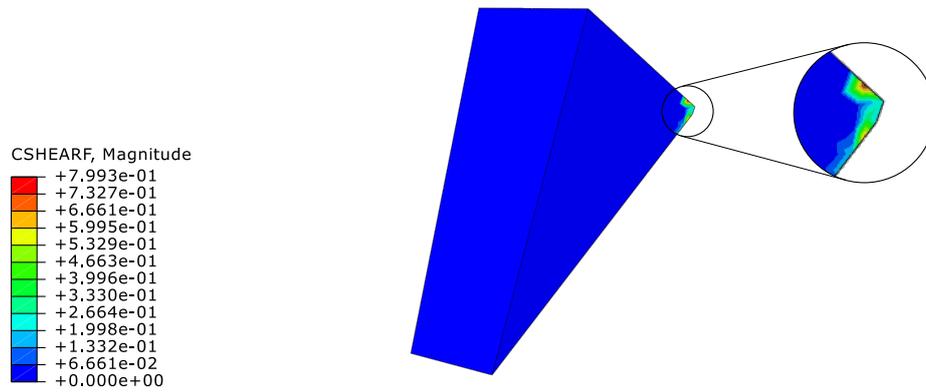


FIGURE 5.46 – Les blocs A, B, C et D se trouvent au niveau de cercles de traction mis en évidence dans une modélisation coque.

Malgré la bonne précision d'impression, les dimensions des blocs imprimés sont légèrement différentes de la géométrie parfaite de la modélisation 3D. La Fig. 5.51 explique cette différence. La géométrie contient des angles très aigus pouvant atteindre 10° et, lors de l'impression, l'épaisseur de couche est de 0.3 mm. Pour $\theta = 10^\circ$ et $e = 0.3$ mm, a vaut alors 1.7 mm. La taille des blocs étant de l'ordre de la dizaine de centimètre, cette erreur – qui a lieu au niveau des arêtes des angles aigus – n'est pas négligeable. Des légers détrompeurs ont alors été ajoutés à la géométrie des blocs (Fig. 5.50), leur rôle principal étant de pouvoir placer les blocs correctement. Ces détrompeurs ne jouent pas le rôle de clé de blocage, néanmoins on peut considérer qu'ils limitent la ruine par glissement des blocs ou plus précisément la ruine par des cinématiques non-heymaniennes (cf. Sec. 3.2.1). L'assemblage du prototype est illustré Fig. 5.52 et Fig. 5.53. Il ne nécessite pas de cintre. On notera la présence d'ouvertures liées aux imperfections dimensionnelles des blocs. À cause de ces imperfections géométriques et des jeux créés, nous ne pouvons pas comparer la flèche maximale sous poids propre à celle de la simulation numérique. Elle est ici de quelques millimètres par rapport à la géométrie idéale non chargée. Cependant, nous avons pu ainsi construire une structure en porte-à-faux, sans utiliser de cintres alors que des anneaux de traction sont présents.

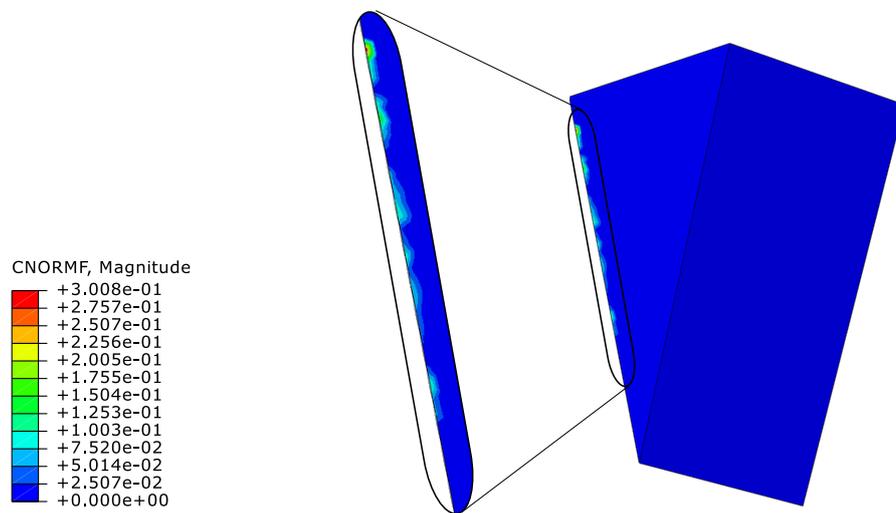


(a) Composantes normales à la surface (N)

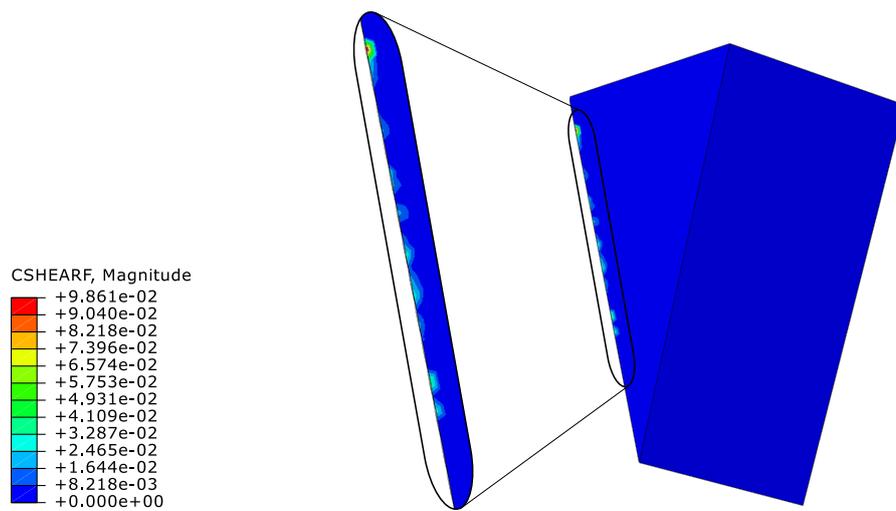


(b) Composantes tangentielles à la surface (N)

FIGURE 5.47 – Efforts de contact localisés entre les blocs A et B. Ici, le bloc A est représenté.



(a) Composantes normales à la surface (N)



(b) Composantes tangentielles à la surface (N)

FIGURE 5.48 – Efforts de contact entre les blocs C et D. Ici, le bloc D est représenté. Le contact est localisé au niveau de l'arête.



FIGURE 5.49 – Le contact n'est pas plan-plan.

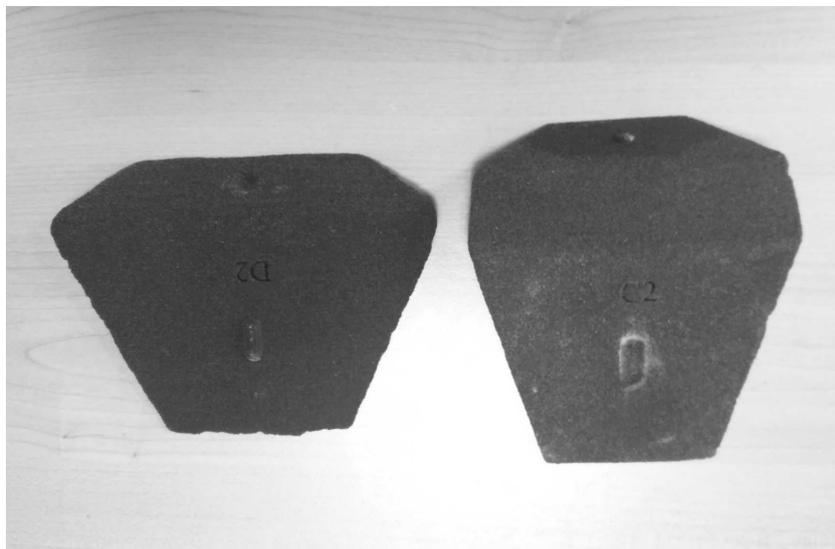


FIGURE 5.50 – Détrompeurs. Ils ne sont présents que pour aligner un bloc avec son voisin du dessous. Ici, la face D2 va sur la face C2. Il n'y a pas de détrompeurs pour aligner les blocs avec les voisins de gauche et de droite.

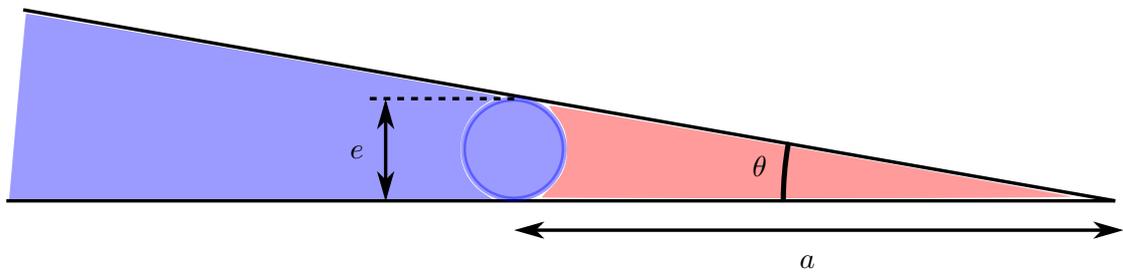


FIGURE 5.51 – En bleu la partie imprimée et en rouge celle non imprimée à cause de l'épaisseur de couche minimum e . a vaut $\frac{e}{\tan \theta}$ et peut atteindre quelques millimètres pour des valeurs de θ petites.



(a)



(b)



(c)



(d)



(e)



(f)

FIGURE 5.52 – Corolle en construction. La base en bois fait 60 cm de long.

5.4 Conclusion du chapitre et ouverture

Une généralisation de la procédure *moving cross-section* a été présentée dans ce chapitre. Celle-ci permet de générer des assemblages topologiquement autobloquants à



FIGURE 5.53 – Corolle finie

partir d'un maillage de quadrangles plans d'une surface non nécessairement plane. Les quadrangles de ces maillages ne sont pas forcément réguliers ; il faut alors vérifier certaines inégalités pour s'assurer du blocage topologique. La notion d'assemblage fortement topologiquement autobloquant a également été introduite. Cette généralisation permet alors de concevoir des structures maçonnées courbes dont la coupe des joints ne suit pas nécessairement le principe de normalité. Il n'est pas nécessaire que les assemblages obtenus de cette manière soient autobloquants, mais en l'étant, on s'assure qu'il n'existe pas de cinématique possible pour extraire un bloc, *dans le sens* du chargement (par exemple du poids).

Cette procédure généralisée nous a permis de construire deux prototypes dont l'un contient des blocs en porte-à-faux et est constructible sans aucun cintre. Cet avantage peut être utile dans une démarche Robot Oriented Design (ROD), introduite en Sec. 1. Néanmoins les blocs obtenus présentent quelques désavantages pour une construction robotisée. Premièrement, ils sont localement très fins et donc fragiles si construits en pierre. De plus, leur forme n'est pas optimisée pour une préhension par un robot. Nous avons développé et sommes en train d'étudier un nouveau type de bloc qui se base sur le principe de ce prototype ; ces blocs étant moins fragiles et plus facilement saisissables par un robot. Ils permettront a priori de construire un plafond sans utilisation de coffrage et de manière robotisée.

Chapitre 6

Localisation relative d'un bloc par rapport à un robot

Résumé : Dans ce chapitre, nous tentons de répondre au problème de vision artificielle robuste pour un environnement de chantier. Nous abordons ce problème par l'estimation de pose relative d'un bloc parallélépipédique par rapport à un bras robot, à partir d'images issues de simples caméras 2D ne nécessitant pas d'étape de calibration. Nous montrons que la localisation peut être effectuée avec une précision millimétrique, sans utiliser aucune images réelles lors de la phase d'apprentissage. En effet, notre approche repose sur une procédure en trois étapes dans laquelle trois réseaux de neurones convolutifs sont entraînés sur une base de données composée uniquement d'images synthétiques. Afin d'évaluer notre approche quantitativement et la rendre comparable aux approches alternatives, nous avons également construit une base de données, constituée d'images réelles contenant un robot et un bloc précisément localisé et orienté.

6.1 Abstraction de la tâche

Localiser pour saisir est une tâche robotique difficile à évaluer quantitativement. Dans cette section, nous expliquons d’abord comment nous exprimons cette tâche en tant qu’une procédure en trois étapes correspondant à trois sous-tâches de vision artificielle pure ; ce qui permet une évaluation quantitative reproductible. Ensuite, nous formalisons ces sous-tâches.

6.1.1 Description de la tâche

La tâche que nous considérons est une simplification de celle décrite en Section 1.2.2.1. Le but est d’être capable de saisir un bloc parallélépipédique inconnu avec un bras robot en utilisant seulement des images de caméras non calibrées (ni intrinsèquement, ni extrinsèquement). Le bloc repose sur une surface globalement plane, sa position, son orientation et ses dimensions sont inconnues. Le robot et le bloc sont dans le champ de vision de plusieurs caméras ; il en découle que le bloc peut être partiellement ou totalement caché par le robot sur certaines des vues. Pour une saisie du bloc réussie, la position et l’orientation du bloc parallélépipédique *par rapport au robot* doivent être connues précisément. À noter cependant qu’il faut seulement la position et l’orientation *relatives* entre le robot et le bloc ; les poses des caméras ne sont pas nécessaires.

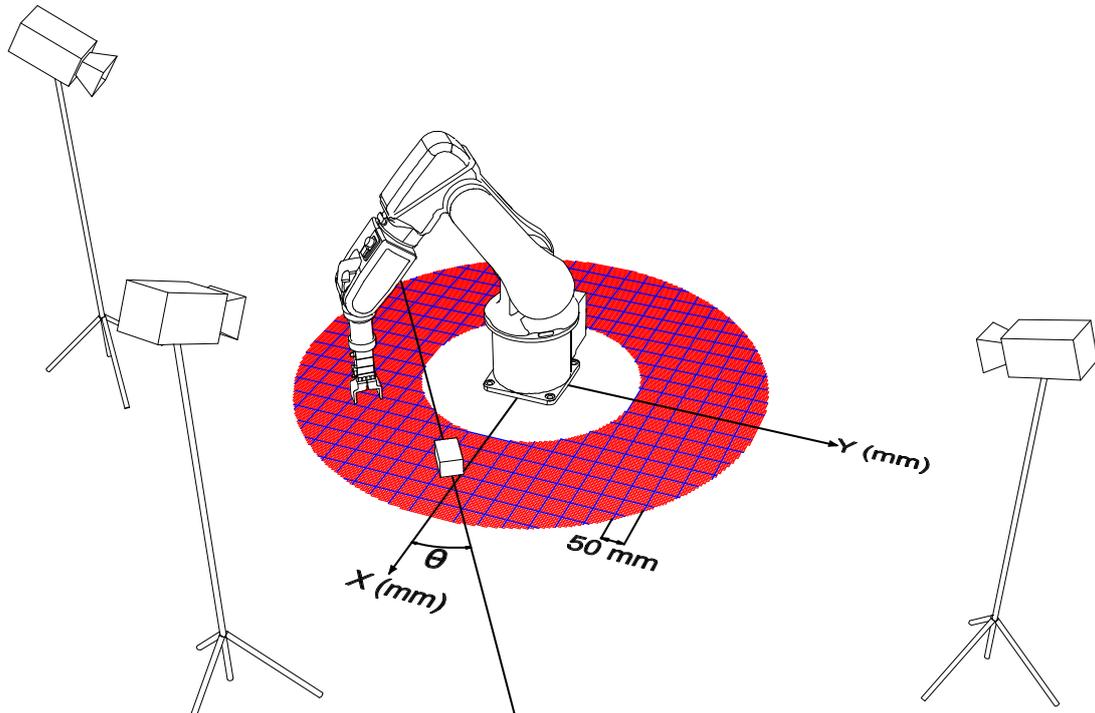
Les seules informations de configuration disponibles pour cette tâche – pouvant éventuellement provenir d’une maquette numérique du bâtiment et des outils de chantier – sont un modèle 3D *non texturé* du robot avec les axes de ses articulations, et une plage de dimensions possibles pour le bloc parallélépipédique. En particulier, l’aspect du robot et celui du bloc sont totalement inconnus. Puisqu’il est difficile de localiser précisément un objet par rapport à un robot, nous avons en fait défini les sous-tâches suivantes, qui correspondent à une procédure de localisation en trois étapes :

1. *Sous-tâche de localisation relative grossière* Nous considérons premièrement le scénario général dans lequel la position et l’orientation du robot et du bloc sont aléatoires ainsi que les valeurs d’angles des différentes articulations du robot. Les caméras, dont les poses et caractéristiques sont aléatoires, fournissent seulement des aperçus de la scène. La sous-tâche est ici d’estimer (grossièrement) la pose du bloc relativement à la base du robot.
2. *Sous-tâche de localisation 2D du préhenseur* Une fois que la position du bloc est ainsi estimée, bien qu’avec une précision et confiance modérées, nous supposons que le préhenseur du robot est déplacé au dessus de cette position prédite grossièrement. Dans ce scénario, le robot et le bloc reste à des positions et avec des orientations aléatoires, mais le préhenseur est situé à une position aléatoire proche du bloc, orienté vers le bas et près à saisir. La seconde sous-tâche est alors de détecter le préhenseur dans l’image afin de permettre de réaliser un zoom pour pouvoir, dans un dernier temps, estimer la pose plus précisément.
3. *Sous-tâche de localisation relative précise* Enfin, en utilisant les zooms obtenus (qui sont en fait des rognages (*crop*) des vues des caméras, centrés sur la zone d’intérêt),

la troisième sous-tâche est d'estimer finement la localisation et l'orientation du bloc par rapport au préhenseur, et donc par rapport au robot, permettant ainsi la saisie. (Ce sont des zooms au sens où la première tâche travaille généralement sur des images de résolution réduite pour des questions d'efficacité.)

À noter que dans ce chapitre, nous nous concentrons sur l'estimation de position dans le plan horizontal sur lequel se tient le robot et supposons que le bloc reste à peu près à plat sur ce plan, avec pas ou peu de variations de hauteur ou d'inclinaison. Cela correspond à une hypothèse réaliste lorsque l'on travaille sur une surface approximativement plane. Adapter la démarche pour pouvoir s'accommoder de variations de hauteur ou d'inclinaison significatives sera l'objet de travaux futurs.

Nous décrivons dans la Sect. 6.3.1 une base de données pour évaluer des méthodes qui abordent ces trois sous-tâches. Elle est constituée d'images réelles d'un robot et d'un bloc ainsi que des annotations précises de la position et orientation relatives du bloc par rapport au robot. Cela permet d'évaluer quantitativement le succès d'une méthode pour une tâche donnée. Cette tâche composite résume en fait une tâche plus générale de positionnement relatif d'objet par rapport à un robot, basée sur des images, dans laquelle les mouvements du robot sont arbitraires. Cependant, puisqu'un tel asservissement visuel est un processus dynamique impliquant des dispositifs réels, il est pratiquement impossible à reproduire synthétiquement et il n'est alors pas possible de comparer deux méthodes données. Notre tâche, constituée de trois étapes de localisation, est une formulation discrète de ce processus dynamique, avec un mouvement intermédiaire prédéfini plausible. L'intérêt est que la tâche est statique et déterministe ; des méthodes concurrentes pouvant ainsi être évaluées et comparées quantitativement et avec sens. Les expériences présentées dans les sections suivantes montrent aussi qu'une telle procédure en trois étapes a du sens pour le contrôle du robot puisque des informations de position et d'orientation plus précises ont pu être obtenues grâce à la sous-tâche de raffinement. De plus, nous avons considéré deux variantes de cette tâche : mono-vue ou multi-vues. En fait, comme le bloc peut parfois être occulté dans une unique vue ou être vu sous un angle peu favorable pour estimer certaines dimensions avec précision, utiliser plusieurs vues fournit plus de robustesse car, ainsi, un bloc a peu de chances d'être caché dans toutes les vues. Cette approche multi-vues permet aussi d'obtenir une plus grande précision en exploitant les informations de chaque vue.



(a) Localisation relative comme un problème de classification dans une grille. Ici avec des cases de 5 mm (grille rouge) pour l'estimation grossière, avec des carrés de 5 cm superposés (grille bleue), pour donner uniquement une idée visuelle des dimensions de l'ensemble.



(b) Image d'apprentissage



(c) Image de validation

FIGURE 6.1 – Nous avons formulé le positionnement relatif d'un bloc par rapport à un robot, en utilisant des caméras non calibrées, comme un problème de classification où l'on prédit la position (x, y) du centre du bloc par rapport à la base du robot, et l'angle θ entre les axes principaux du bloc et du robot (a). Nous montrons que nous pouvons entraîner un CNN sur des images synthétiques pour réaliser cette tâche, en utilisant des poses, des apparences et des paramètres de caméras tous aléatoires (b), et ensuite l'utiliser pour réaliser des prédictions très précises de positionnement relatif sur des images réelles (c).

6.1.2 Robot et référentiel

Nous avons besoin de définir un référentiel afin de formaliser les sous-tâches de positionnement. Puisque les blocs sont symétriques (invariant par une rotation de 180° dans le plan horizontal), et puisque le problème est principalement centré sur le robot, nous avons décidé d'associer le référentiel au robot. Pour définir les axes du repère avec sens, le robot doit avoir des asymétries. Plutôt que d'apposer des marqueurs visuels sur le robot, qui ne sont pas robustes, spécifiquement dans le cas d'un chantier avec de la poussière, de la boue et possiblement du mauvais temps, nous avons décidé de ne compter seulement sur les asymétries de la forme intrinsèque du robot. Concrètement, dans la base de donnée que nous avons créée pour cette tâche (voir Sect. 6.3.1), nous avons utilisé un robot (modèle IRB120 de l'entreprise ABB) avec les principales caractéristiques de forme suivantes :

- La base du robot est globalement cylindrique. Cependant, elle est fixée à son support avec des écrous par l'intermédiaire d'une plaque approximativement carrée dont les orientations peuvent être utilisées pour définir les axes X et Y . De plus, les câbles sont connectés au robot via un boîte cubique accolée à la base du cylindre, apportant ainsi une asymétrie que nous utilisons pour définir l'axe X ; l'axe Y est alors défini orthogonalement et orienté par convention. L'orientation relative du bloc θ est l'angle entre la plus grande dimension du bloc (sur le plan) et l'axe X . Puisqu'ici nos blocs parallélépipédiques sont symétriques, cet angle est seulement défini entre 0° et 180° . Ce référentiel est illustré sur la Fig. 6.1a et est utilisé pour la sous-tâche de localisation grossière.
- Juste au dessus des mors du préhenseur, une petite vis, ainsi qu'un petit cache carré en plastique, rendent le préhenseur asymétrique, comme illustré Fig. 6.2. Comme décrit ci-dessus, cela permet de définir deux axes X et Y et l'angle θ de l'axe principal du bloc par rapport à l'axe X . Ce référentiel est relatif au préhenseur plutôt qu'à la base du robot, mais on peut immédiatement le relier au référentiel du robot puisque les valeurs des joints du robots ainsi que la position du préhenseur sont connues. À noter que ce second référentiel a un sens seulement si le préhenseur est orienté verticalement (vers le bas). Il est utilisé pour la sous-tâche de localisation précise.

De telles asymétries de forme peuvent facilement être rajouté à n'importe quel robot si elles n'existent pas déjà.

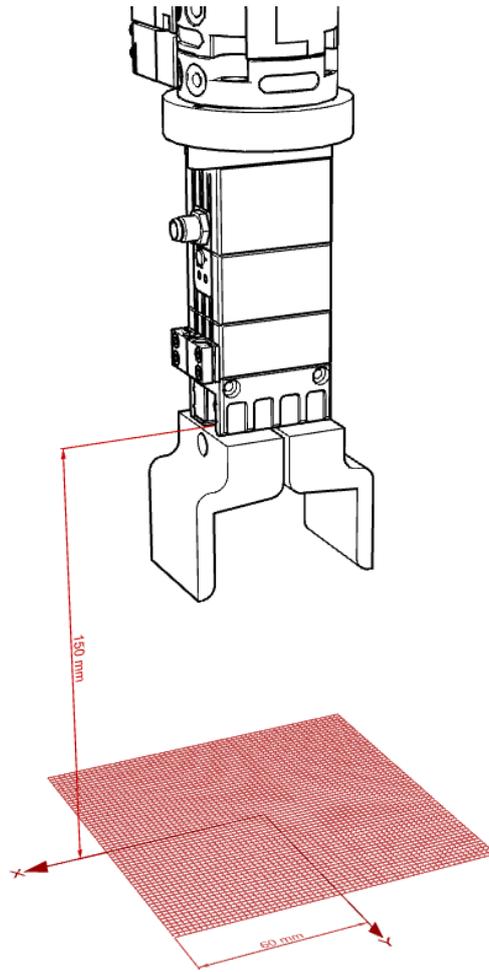


FIGURE 6.2 – Référentiel pour l’estimation de localisation précise. Noter l’asymétrie entre le côté gauche et le côté droit du préhenseur. Dans nos expériences de classification, pour estimer la position du bloc, nous avons utilisé une grille de 12 cm avec des pas de 2 mm.

6.2 Résolution de la tâche avec des CNNs et des images synthétiques

Retrouver la position exacte d’un objet (robot, pièces à attraper, obstacles, etc) à partir d’une unique image et d’un modèle 3D est un problème difficile, spécialement quand la texture de l’objet est inconnue ou peu fiable et quand ses déformations (articulations du robot, variantes de taille et de forme de l’objet) le font varier drastiquement. Pourtant, à partir de plusieurs vues, les caméras peuvent être calibrées et une reconstruction 3D peut être réalisée, permettant au problème d’être résolu après un alignement du modèle 3D. Cependant, la calibration multi-vues et la reconstruction peuvent être des problèmes difficiles, nécessitant un temps de traitement significatif possiblement in-

compatible avec la vitesse attendue d’un robot sur un chantier. Il en va de même de l’alignement des modèles 3D sur un nuage de points. Ici, nous tentons plutôt de résoudre le problème de manière plus directe en entraînant un CNN à localiser avec fiabilité un bloc par rapport au robot, grâce à une seule image. Si plusieurs images issues de plusieurs caméras sont disponibles, les estimations de localisation peuvent être agrégées pour améliorer la précision globale. De plus, nous n’utilisons que des images synthétiques pour l’apprentissage, pour éviter le coût élevé de création et d’annotations de données réelles. Plus précisément, nous entraînons trois CNNs, un pour chaque sous-tâche. Le premier réseau estime la localisation approximative du bloc par rapport à la base du robot dans une unique image. Une fois que le préhenseur du robot a été déplacé au dessus de cet emplacement approximatif, le deuxième réseau localise le préhenseur dans l’image dans cette nouvelle configuration. Enfin, en utilisant un rognage de l’image autour de l’estimation de position du préhenseur, le troisième réseau estime la pose précise du bloc par rapport au préhenseur et donc par rapport au robot. Bien qu’il y a plusieurs étapes dans cette approche, elle diffère d’une chaîne de traitement avec calibration de caméras, reconstruction 3D et alignement du modèle, dans le sens que les erreurs ne s’accumulent pas. En effet, en supposant que le premier et le deuxième réseaux sont suffisamment fiables de manière à obtenir une image zoomée (en fait, un rognage) du préhenseur avec le bloc, le troisième et dernier réseau peut effectuer une estimation précise qui ne dépend pas de la précision des sous-tâches précédentes, y compris le centrage du rognage.

6.2.1 Estimation de pose relative avec un CNN

Approche par classification Puisque la position (x, y) aussi bien que l’angle θ sont des quantités continues, il est naturel de formuler leur estimation comme un problème de régression. Cependant, une telle formulation n’est pas adaptée pour représenter les ambiguïtés. Par exemple, puisque le bras robot est presque symétrique, il serait naturel pour notre système d’hésiter entre deux positions symétriques durant l’étape d’estimation fine. Dans une formulation par régression, il n’y a pas de façon naturelle de gérer de telles ambiguïtés. Au contraire, si on prédit des probabilités pour chaque position du bloc, une distribution multi-modale peut être prédite dans les cas ambigus. Pour cette raison, nous avons discrétisé l’espace des poses possibles et prédisons des probabilités pour chaque case (*bin*). Il a été montré qu’une telle formulation était plus efficace dans des problèmes similaires, tels que l’estimation de position de points clés et l’estimation d’orientation (Tulsiani and Malik, 2015; Su et al., 2015; Massa et al., 2016a). Concrètement, étant donné le niveau de précision que l’on cherche à atteindre (voir Sect. 6.3.2), l’estimation de pose grossière utilise des cases¹ carrées de taille 5 mm et l’estimation précise des cases de taille 2 mm.

1. Dans ce travail, nous faisons une distinction entre cases et classes. Les cases correspondent ici à des coordonnées discrètes (x, y) . L’architecture du réseau de classification que nous utilisons (voir plus bas) est telle que celui-ci prédit x et y séparément. Ainsi, par exemple, la classe x_0 correspond à l’ensemble de cases (x_0, y) pour y quelconque. Les cases sont donc des carrés et les classes sont ici un ensemble de cases.

Architecture du réseau Nous avons résolu le problème de classification que l'on vient de définir en entraînant un CNN. Plus précisément, nous avons utilisé un réseau ResNet-18, entraîné à partir de zéro, et une perte (standard) qui mesure l'entropie croisée. En effet, ce type de réseau a montré de bonnes performances, notamment sur des tâches de classification, tout en restant relativement léger. On a utilisé une taille de paquet (*batch size*) de 128, un coefficient de diminution des pondérations (*weight decay*) de 10^{-4} et une inertie (*momentum*) de 0.9. On a entraîné notre modèle avec un taux d'apprentissage (*learning rate*) initial de 10^{-2} jusqu'à convergence (250 itérations pour l'estimation grossière, 11 itérations pour la détection du préhenseur et 186 itérations pour l'estimation précise), puis avec un taux d'apprentissage de 5.10^{-3} pendant 15 itérations (pour l'estimation grossière et précise). Nous n'avons pas étudié en détail l'influence de la taille du réseau, mais on peut raisonnablement espérer un léger gain de performance en l'optimisant, c'est-à-dire en utilisant un réseau plus profond ou plus large, tout en le gardant assez petit pour éviter le surapprentissage.

Cependant, la définition de la forme de la sortie du réseau n'est pas évidente. Définir simplement une classe par case mènerait à trop de classes. Par exemple, il y a plus de 17500 cases de positions 2D (x, y) atteignables pour l'estimation grossière (3600 pour l'estimation précise), chacune d'elles pouvant être séparées en 36 cases pour l'orientation (90 pour l'estimation précise). Cela ne mènerait pas seulement à une utilisation de la mémoire beaucoup plus grande, mais cela nécessiterait aussi plus de données d'apprentissage. En effet, nous avons montré dans nos premières expériences (voir Sect. 6.3.3) que, malgré le partage d'information intuitif qui pourrait être utilisé pour prédire les différentes classes, 80 images par classe de localisation (pour un total de plus de 280k images d'apprentissage) ne sont pas assez pour éviter le surapprentissage. L'alternative la plus simple serait de simplement entraîner trois réseaux indépendants pour prédire x , y et l'angle θ séparément (Fig. 6.3b). Cependant, nous montrons qu'il vaut mieux entraîner un réseau multi-tâches qui prédit des probabilités pour x , y et θ indépendamment mais calcule une unique représentation dans toutes les couches exceptée la dernière – autrement dit, les couches cachées du modèle sont partagées mais pas la dernière couche (Fig. 6.3c). On remarque que, contrairement à la prédiction du réseau avec une probabilité par position 2D, ce réseau ne souffre pas du phénomène de surapprentissage..

Prédiction multi-vues Réaliser une classification au lieu d'une régression a aussi l'avantage de donner une façon naturelle de fusionner les prédictions de plusieurs vues. En effet, la sortie du réseau peut être interprétée comme des log-probabilités. Si l'on considère les informations des différentes vues comme indépendantes, la probabilité de position connaissant toutes les images est simplement le produit des probabilités individuelles. On peut donc simplement prédire le maximum de la somme des sorties du réseau appliqué aux différentes images. En fait, l'estimation à partir d'une unique vue peut être ambiguë, par exemple parce que la dimension exacte du bloc est inconnue ou parce que le bloc est caché par le robot. Fusionner les prédictions de plusieurs vues est crucial au succès de notre approche comme montré en Sect. 6.3.5. À noter que non seulement cette approche s'adapte bien avec un nombre grandissant de caméras mais aussi, à l'inverse,

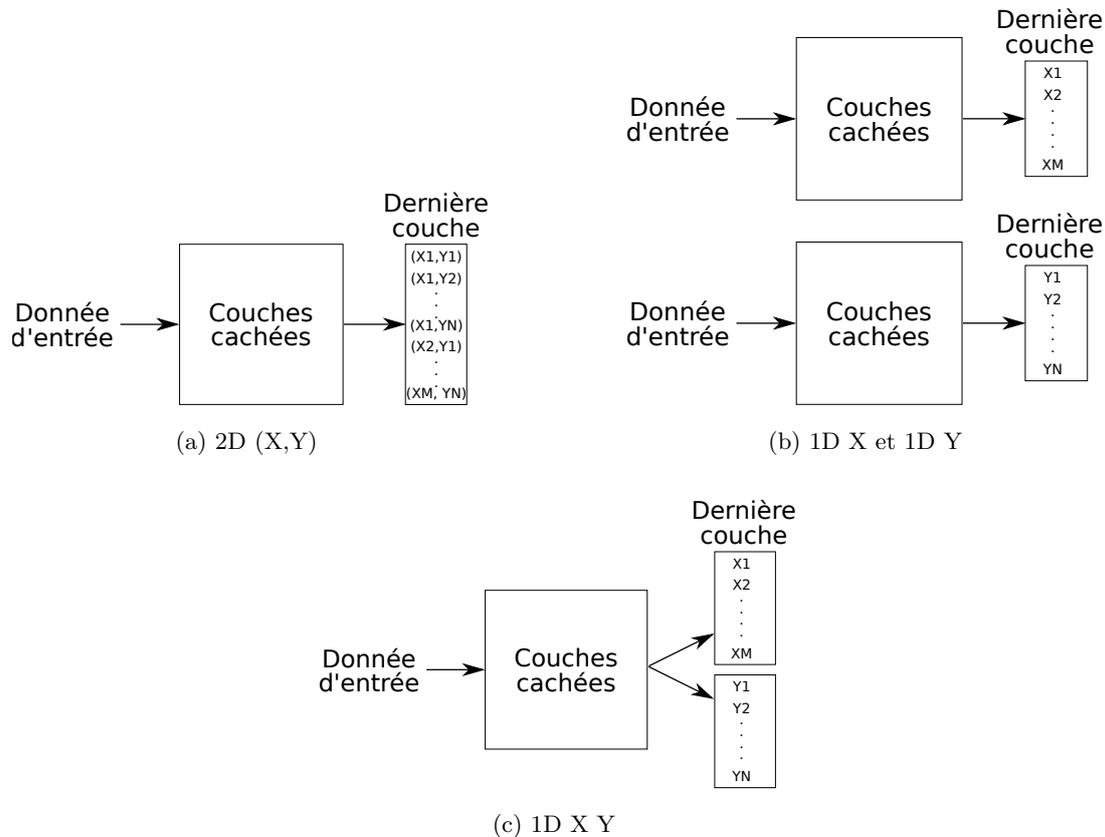


FIGURE 6.3 – Différentes architectures possibles pour la sortie du réseau. Nous montrons qu’il vaut mieux utiliser l’architecture (c).

permet une dégradation douce (allant même jusqu’au cas à une seule vue) dans le cas où certaines caméras sont déplacées, inopérantes ou tournées et visant en dehors de la scène. Des stratégies plus compliquées peuvent bien sur être utilisées, telles qu’entraîner un réseau à accepter directement plusieurs vues en entrée ou entraîner un réseau de neurones récurrents qui "verrait" les différentes vues une par une.

6.2.2 Base de données d’apprentissage synthétique

La création d’une grande base de données d’apprentissage constituée de photographies réelles, et avec des variations représentatives de celles rencontrées pendant les scénarios de test réels, est long et difficile. Notamment lorsque l’on considère la variabilité propre à un environnement non contrôlé, c’est-à-dire lié à la poussière, saleté ou l’éclairage naturel. À la place, nous avons généré trois ensembles d’images synthétiques ainsi que les vérités terrains des poses associées que l’on utilise pour l’apprentissage et pour la validation de chacune des sous-tâches précédemment mentionnées. Plutôt que

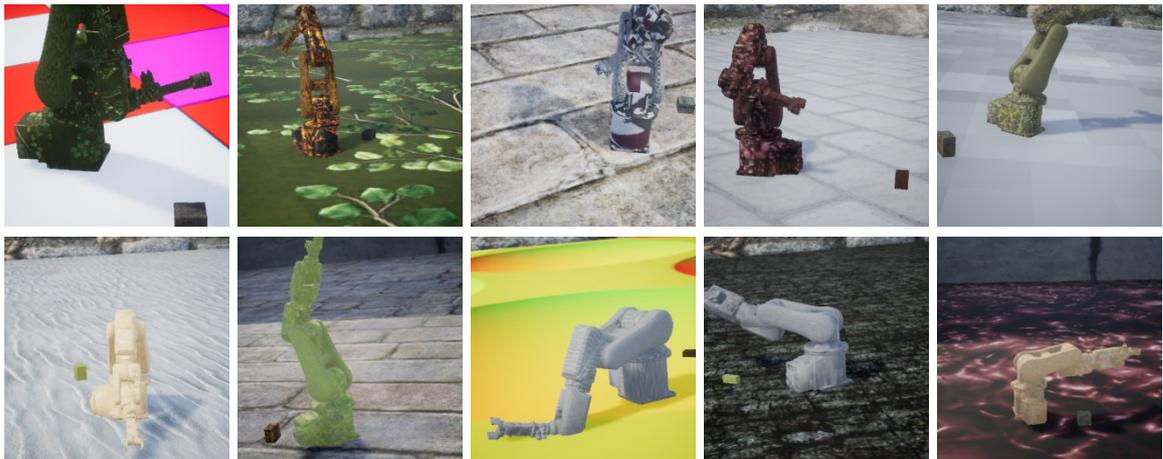
de passer beaucoup de temps à générer des images photoréalistes ou utiliser une forme d'adaptation de domaine (voir Sect. ??), nous avons choisi d'appliquer une stratégie de génération massive d'images (des centaines de milliers d'images non photoréalistes) avec une randomisation massive ciblant les propriétés spécifiques que l'on souhaite renforcer. Dans cette base de données, comme illustré sur la Fig. 6.4, le bras robot est placé sur un sol plat, et un bloc parallélépipédique est posé à plat à ses côtés, avec les configurations suivantes :

1. Le robot et le bloc dans des poses aléatoires pour l'estimation grossière,
2. Le robot avec le préhenseur dans une pose verticale aléatoire pour la détection 2D du préhenseur,
3. Des zooms sur le préhenseur vertical avec un bloc placé aléatoirement à ses alentours pour l'estimation fine.

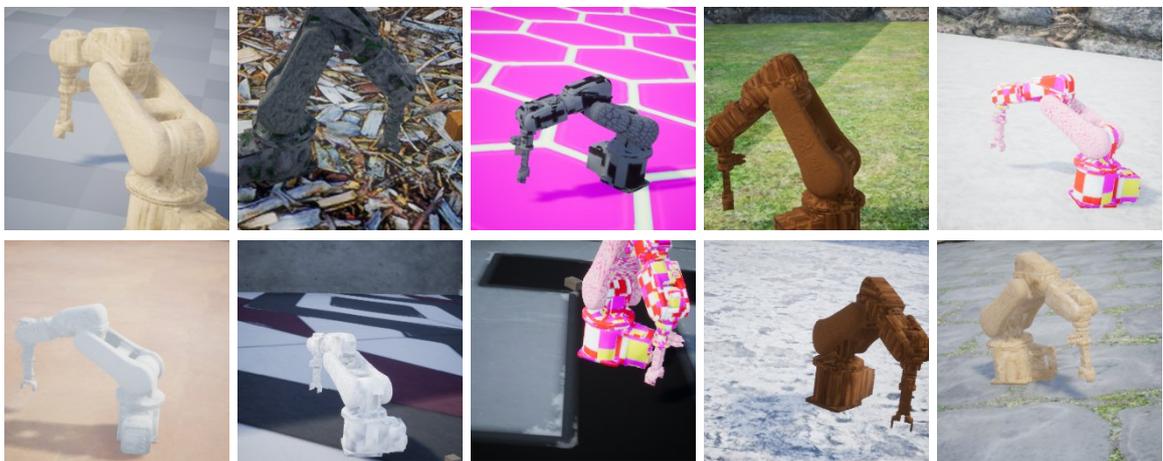
Pour encourager le modèle à généraliser autant que possible, nous avons introduit la randomisation suivante dans la génération des images synthétiques :

- Orientation et position de la base du robot, angles des articulations (joints) du robot,
- Dimensions, orientation et position du bloc parallélépipédique,
- Textures pour le sol, le robot et le bloc,
- Centre, cible, rotation et distance focale de la caméra.

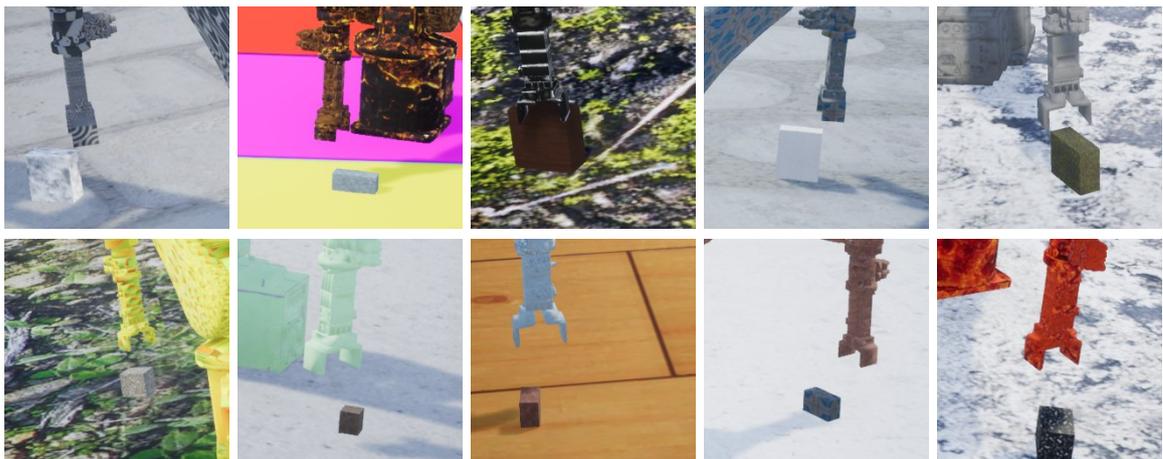
Des détails sur cette base de données synthétiques sont donnés en Annexe A.



(a) Exemples de données d'apprentissage pour l'estimation grossière (du bloc par rapport à la base du robot)



(b) Exemples de données d'apprentissage pour la détection 2D du préhenseur



(c) Exemples de données d'apprentissage pour l'estimation de pose précise (du bloc par rapport au préhenseur)

FIGURE 6.4 – Notre base de données d'apprentissage composée d'images synthétiques couvre une grande variété de points de vue différents, d'angles d'articulations, de poses et de dimensions de bloc, et de textures (pour le robot, le bloc et l'environnement), rendant le modèle entraîné robuste à des changements de chacun de ces paramètres.

6.3 Résultats expérimentaux et analyse

Nous décrivons dans cette section une base de données d'évaluation et l'utilisons pour analyser en détail les performances de notre approche d'apprentissage directe, en incluant les différents choix de conception. Cette base de données d'évaluation, composée de photographies réelles, est différente de celle de validation mentionnée en Sec. 4.1.4.2. Cette dernière, composée d'images synthétiques, permet seulement de vérifier que le réseau a bien appris sur les données synthétiques et qu'il n'y a pas eu de surapprentissage, alors que la base de données d'évaluation que nous décrivons ici est composée de photographies réelles et permet de vérifier que le réseau entraîné sur des images synthétiques fonctionnent bien sur des images réelles.

6.3.1 Base de données d'évaluation

Pour autant que nous le sachions, il n'existe pas de base de données concernant la tâche de localisation relative robot-objet de haute précision. Nous avons créé une telle base de données, avec des images réelles, pour la tâche composite décrite Sect. 6.1. À noter qu'il s'agit seulement d'une base de données de test (d'évaluation) et non d'une base de données d'entraînement. Cette base de données est divisée en trois parties, une par sous-tâche :

1. Étant donnée une image du robot et du bloc, trouver leur pose relative grossière dans le plan support,
2. Étant donnée une image du robot avec le préhenseur vertical pointant vers le bas, trouver la position du préhenseur dans l'image,
3. Étant donnée une image zoomée du préhenseur vertical pointant vers le bas et d'un bloc, trouver leur pose relative précise dans le plan support.

Les poses relatives du bloc sont cohérentes dans les trois parties de la base de données ; pour chaque pose aléatoire du bloc, nous prenons en photo :

- (i) Une vue de loin de la scène où les joints du robot sont avec des valeurs d'angle aléatoires,
- (ii) Une vue de loin de la scène avec le préhenseur déplacé près du bloc, vertical et pointant vers le bas,
- (iii) Un rognage de cette même image, plus ou moins centré sur le préhenseur et assez large pour montrer aussi le bloc.

Pour étudier les bénéfices d'avoir plusieurs vues de la même scène, chaque configuration est en fait vue et enregistrée depuis trois caméras. De plus, nous avons enregistré pour chaque configuration du type (ii)-(iii) une pose supplémentaire avec le préhenseur tourné de 90° autour de l'axe vertical (voir Fig. 6.5).

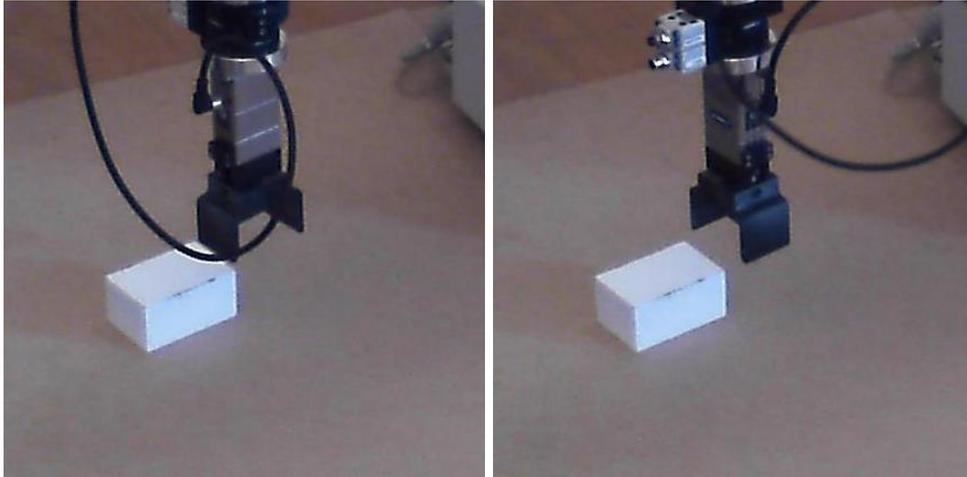


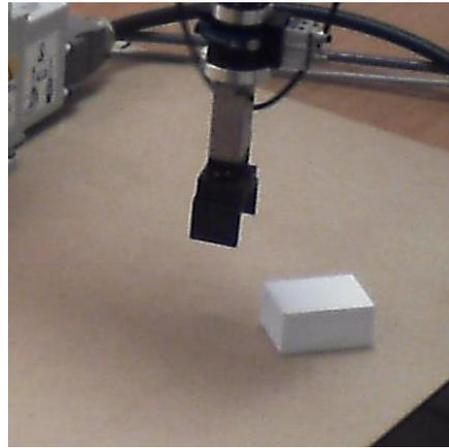
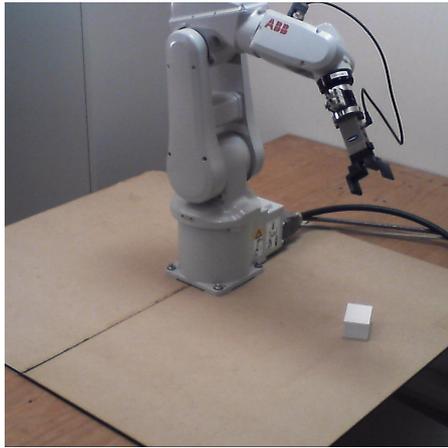
FIGURE 6.5 – Exemple du préhenseur approximativement placé au dessus du bloc. Notons les deux positions de ce préhenseur (P1 et P2) utilisées pour éventuellement améliorer la robustesse de l’estimation de pose du bloc.

De plus, nous avons fait trois variantes de chacune de ces (sous) bases de données, correspondant à des difficultés d’environnement différents, comme illustré sur la Fig. 6.6 :

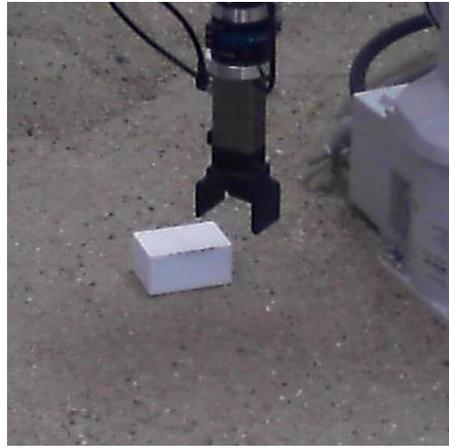
- (a) Une base de données en condition de laboratoire (‘lab’) dans laquelle le robot et le bloc sont sur une table plane sans distracteur, ni texture,
- (b) Une base de données dans des conditions plus réalistes (‘terrain’), dans laquelle la table est couverte de terre, de sable et de graviers, rendant la surface légèrement non plane avec des blocs qui ne reposent pas parfaitement horizontalement, et rendant l’apparence plus proche de ce que l’on pourrait voir sur un chantier,
- (c) Une base de données défavorable (‘défav’) dans laquelle la table est couverte de bouts de papiers qui agissent comme des distracteurs car ils peuvent aisément être confondus avec des blocs parallélépipédiques.

La base de données dans sa totalité couvre environ 1300 poses (576 pour ‘lab’, 639 pour ‘défav’ et 114 pour ‘terrain’), vues depuis trois caméras, avec une pose de préhenseur pour la configuration (i) et deux poses de préhenseur pour les configurations (ii)-(iii), aboutissant à environ 11700 images annotées. Plus de détails sur cette base de données d’évaluation² sont disponibles en Annexe A. Dans ce qui suit, pour étudier différentes propriétés et choix de conception de notre approche, nous reportons les résultats pour la base de données ‘lab’ uniquement, sauf mention contraire.

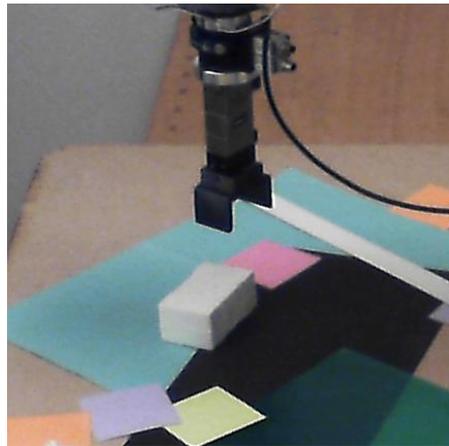
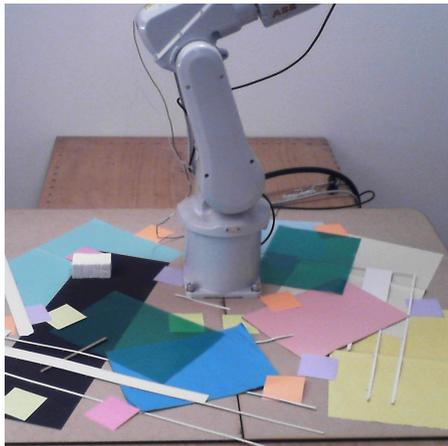
2. Cette base de données est publiquement disponible sur la page du projet <http://imagine.enpc.fr/~loingvi/unloc/>



(a) Base de données 'lab'.



(b) Base de données 'terrain'.



(c) Base de données 'défav'.

FIGURE 6.6 – Exemples d'images pour les trois variantes de notre base de données d'évaluation, pour l'estimation grossière (gauche) et l'estimation précise (droite) : conditions de laboratoire propres (a), conditions plus proche d'un terrain avec des graviers (b) et conditions plus défavorables avec des distracteurs (c).

6.3.2 Métrique d'évaluation

Pour évaluer les résultats d'estimation de pose x , y , θ , plusieurs mesures peuvent être considérées. La plus naturelle est de mesurer la *précision*, en calculant la moyenne et l'écart type des erreurs d'estimation e_x , e_y , e_θ . Bien que cela produise des valeurs aisément interprétables, ce n'est pas une mesure utile de la capacité de la méthode à fournir une prédiction assez précise (pour la préhension) puisqu'elle ne prend pas en compte une erreur possible maximale. Nous présentons aussi des mesures du *succès* en terme de taux d'erreurs d'estimation e_x , e_y , e_θ qui sont sous des seuils donnés, en l'occurrence, qui sont assez précises pour que la saisie du bloc soit réussie avec le préhenseur utilisé.

Concrètement, pour le réseau d'estimation précise, nous avons considéré un domaine carré de 12 cm de côté, comme illustré sur Fig. 6.2. L'estimation précise a alors du sens si l'erreur de prédiction de l'estimation grossière qui précède est inférieure à 6 cm. En outre, étant donnée l'ouverture des mors du préhenseur que l'on utilise, pour qu'une saisie soit réussie, la précision requise est de 5 mm pour x , 5 mm pour y et 2° pour l'orientation du bloc. Concernant l'estimation de position 2D du préhenseur dans l'image (deuxième sous-tâche intermédiaire), nous n'avons pas créé de vérité terrain ; nous aurions dû pour cela effectuer une calibration du robot et des caméras. Pour nos expériences, nous avons à la place vérifié manuellement toutes les prédictions de position du centre du préhenseur et compté le nombre de fois où cette prédiction était en dehors de la boîte englobante (*bounding box*) du préhenseur dans l'image.

6.3.3 Architecture du réseau

Comme mentionné en Sect. 6.2.1, nous avons décidé de prédire indépendamment x , y et θ en utilisant un unique réseau qui calcul une unique représentation sauf pour la dernière couche. Pour valider notre choix, nous avons évalué la précision de la prédiction de la coordonnée x de notre scénario d'estimation grossière avec trois réseaux différents. Ces réseaux prédisent la pose du bloc avec trois approches différentes :

- Le long de l'axe x uniquement, avec des cases de largeur 5 mm,
- Avec des cases carrées (x, y) de taille 5×5 mm,
- Le long des axes x et y séparément, avec des cases de largeur 5 mm, et pour l'orientation θ , avec des cases de taille 5° .

Les résultats sont reportés dans la Table 6.1. Ils montrent clairement que le réseau prédisant la position pour chaque carré 5×5 mm échoue, même si en théorie il pourrait représenter des ambiguïtés plus complexes dans l'estimation de la pose. L'analyse des performances d'apprentissage et de validation sur la base de données synthétiques montre qu'il y a surapprentissage. Au contraire, le réseau prédisant x seulement fonctionne bien, étant capable de généraliser apparemment vis-à-vis de plusieurs positions y . Le réseau unique prédisant séparément x , y et l'angle θ fonctionne encore mieux, montrant un bénéfice net dans un apprentissage joint. Nous avons utilisé cette dernière architecture pour toutes les expériences suivantes. Plus de détails techniques sont donnés en annexe.

TABLE 6.1 – Taux de succès de l’estimation de pose grossière pour trois architectures de réseau différentes (cf. Fig. 6.3), dans une configuration avec une seule caméra, pour la base de données ‘lab’, mesuré comme le pourcentage d’erreurs d’estimation pour x inférieur à 60 mm.

Architecture	cases x 1D	cases (x, y) 2D	cases x, y et θ 1D
% ($e_x \leq 60$ mm)	97.0	6.1	98.7

6.3.4 Apprentissage sur données synthétiques et validation sur données réelles

Nos réseaux sont entraînés sur des données synthétiques uniquement. Une des premières questions à se poser est de savoir comment ils se comportent sur des images réelles. Dans cette section, nous comparons les performances de notre réseau pour l’estimation précise sur les données synthétiques et sur les données réelles d’évaluation. Les résultats pour une seule caméra et une seule orientation du préhenseur sont fournis dans la Table 6.2.

TABLE 6.2 – Taux de succès pour l’estimation de pose précise sur la base de données synthétiques de validation et sur la base de données réelles d’évaluation, pour une caméra et une orientation de préhenseur.

Base de données	synthétique	‘lab’	‘terrain’	‘défav’
% ($e_x \leq 5$ mm)	85.3	62.8	61.1	47.5
% ($e_y \leq 5$ mm)	85.3	67.6	65.2	46.3
% ($e_\theta \leq 2^\circ$)	68.4	83.2	78.2	61.0

La différence de succès entre la base de données synthétiques et celle des données ‘lab’ est d’environ 20% pour l’estimation de position (prédiction de x et de y). Étonnamment, le réseau est meilleur pour les bases de données ‘lab’ et ‘terrain’ pour l’estimation d’orientation (prédiction de θ) que sur la base de données synthétiques. Nous l’expliquons par la variété de textures dans les données synthétiques agissant comme des distracteurs alors que dans les données ‘lab’ et ‘terrain’, les arêtes du bloc sont clairement visibles et non ambiguës. De plus, puisque tous les blocs ont des dimensions différentes dans les données synthétiques, parfois l’arête du bloc la plus courte et celle la plus longue sont presque de même longueur, menant à une ambiguïté d’orientation ; alors que les longueur et largeur du bloc dans les données réelles sont significativement différentes. Comme prévu, la base de données ‘défav’ donne des résultats plus mauvais, sans que néanmoins les performances ne s’effondrent complètement.

6.3.5 Agrégation des vues

Comme indiqué en Sect. 6.2.1, les estimations distinctes de plusieurs caméras peuvent être agrégées en une unique estimation. Nous évaluons en détail l'intérêt d'agréger les estimations des vues de différentes caméras, en considérant également l'exploitation simultanée de vues avec deux orientations du préhenseur (tourné de 0° et tourné de 90°) comme montré sur la Fig. 6.5. Nous indiquons nos résultats pour l'estimation précise dans la Table 6.3. Cette table nous permet de faire deux observations importantes. Premièrement, en comparant les trois premières colonnes à la dernière, on remarque que, sans surprise, utiliser plusieurs points de vue améliore les performances. L'amélioration en performance est même relativement substantielle. Deuxièmement, en comparant les deux premières lignes de chaque sous-tableau à la troisième, on remarque qu'agréger les prédictions de deux images issues d'une même caméra mais avec deux orientations de préhenseur (tourné de 0° et 90°) améliore aussi les résultats. C'est à première vue surprenant puisque les deux images sont extrêmement similaires, mais cela peut s'expliquer par le fait que l'extrémité du bras robot est presque symétrique et estimer son orientation peut être difficile pour certaines orientations, comme illustré Fig. 6.5. Une petite incohérence peut être observée dans l'estimation de y (2.6 ± 4.4 mm), qui n'est pas aussi précise et robuste que l'estimation de x (2.3 ± 1.8 mm). Bien qu'en général les différences entre x et y peuvent être expliquées par l'asymétrie de la base du robot (voir Fig. 6.1) et l'ensemble non uniforme des positions des caméras, placées dans le demi-espace positif des x (voir Fig. 6.7), la raison principale ici est en fait que le réseau retourne une estimation totalement fautive (97 mm d'erreur) pour l'une des 576 poses, alors que les erreurs extrêmes sont sinon rares et jamais plus grandes que 15 mm. Cela impacte e_y à la fois dans la moyenne (0.2 mm) et dans l'écart type (plus de 2 mm). À noter que cette seule estimation fautive affecte seulement la précision des mesures, et non le succès de réussite puisqu'elle représente seulement 0.17% des poses. Bien que non décrites ici, des observations similaires peuvent être faites à propos de l'utilisation et de l'agrégation de plusieurs vues pour réaliser l'estimation grossière. Ces résultats positifs valident notre stratégie simple d'agréger les prédictions de chaque caméra pour chaque orientation du préhenseur. Sauf mention contraire, dans ce qui suit nous reportons seulement des résultats agrégés.

6.3.6 Procédure en trois étapes et taux de succès de la saisie

Nous avons présenté une approche en trois étapes pour estimer précisément la pose relative d'un bloc par rapport au robot. Nous examinons ici si cette approche est réaliste en ce qui concerne le taux de succès pour la saisie et si toutes les étapes sont vraiment nécessaires. Pour vérifier si l'approche a du sens, on peut simplement regarder le taux de succès des deux premières étapes de la procédure. Comme on peut le voir dans la Table 6.4, la première étape (c'est-à-dire l'estimation de pose grossière) a un taux de succès de 99.8%. C'est ainsi extrêmement fiable et c'est presque toujours suffisamment précis pour permettre d'estimer ultérieurement une pose plus finement. La seconde étape de la procédure nécessite de localiser le préhenseur dans l'image, pour ensuite rogner

TABLE 6.3 – Précision (moyenne et écart type) pour la tâche d’estimation précise en agrégeant différents points de vue (caméras 1,2,3) et différentes orientations du préhenseur (P1 et P2) sur la base de données ‘lab’.

erreur x (mm)				
Configuration	Caméra 1	Caméra 2	Caméra 3	Toutes les caméras
Préhenseur P1	7.5 ± 16.0	8.2 ± 16.6	10.9 ± 21.7	3.4 ± 7.6
Préhenseur P2	7.2 ± 15.1	6.7 ± 11.0	10.7 ± 22.7	2.9 ± 4.6
P1 et P2	4.4 ± 6.9	4.1 ± 6.2	6.1 ± 14.2	2.3 ± 1.8

erreur y (mm)				
Configuration	Caméra 1	Caméra 2	Caméra 3	Toutes les caméras
Préhenseur P1	6.7 ± 14.6	7.1 ± 15.6	10.4 ± 21.8	3.0 ± 5.4
Préhenseur P2	6.4 ± 14.5	5.5 ± 11.3	9.8 ± 20.8	2.8 ± 4.3
P1 et P2	4.4 ± 7.7	4.3 ± 6.5	6.5 ± 15.0	2.6 ± 4.4

erreur θ ($^\circ$)				
Configuration	Caméra 1	Caméra 2	Caméra 3	Toutes les caméras
Préhenseur P1	1.4 ± 3.7	1.9 ± 7.8	2.2 ± 8.0	0.8 ± 0.6
Préhenseur P2	1.3 ± 3.6	1.4 ± 3.9	2.2 ± 8.5	0.8 ± 0.6
P1 et P2	1.1 ± 3.4	1.1 ± 4.4	1.3 ± 5.6	0.7 ± 0.6



FIGURE 6.7 – Acquisition d’images réelles pour créer notre base de données d’évaluation, avec trois caméras à des positions arbitraires et changeantes.

TABLE 6.4 – Taux de succès pour l’estimation de pose grossière sur la base de données réelles d’évaluation, avec 3 caméras et une seule orientation de préhenseur.

Base de données	'lab'	'terrain'	'défav'
% ($e_x \leq 60$ mm)	99.8	100.0	84.7
% ($e_y \leq 60$ mm)	100.0	99.1	88.3
% ($e_\theta \leq 10^\circ$)	99.8	99.1	88.7
% ($e_x, e_y \leq 60$ mm)	99.8	99.1	77.5

une région d’intérêt de cette image et effectuer la localisation précise. Puisque la base de données d’évaluation ne contient pas de vérités terrains exactes de la position du préhenseur dans l’image (voir Sect. 6.3.2), nous avons vérifié manuellement toutes les prédictions de notre réseau entraîné pour localiser le centre du préhenseur et compter les cas pour lesquels la prédiction est en dehors de la *bounding box* du préhenseur dans l’image. En ce sens, notre second réseau détecte correctement le préhenseur dans 99.1% des images, ce qui est également un taux de succès élevé. En tout, le taux de succès global des deux premières étapes avant l’estimation fine est de 98.9%, ce qui confirme que ces étapes préliminaires ne dégradent pas significativement l’estimation finale.

Maintenant, pour vérifier si la procédure en trois étapes améliore la précision par rapport à une unique procédure, nous pouvons comparer le pourcentage d’images avec des erreurs inférieures à 5 mm et 2° dans deux configurations : l’une dans laquelle une seule vue large de loin est considérée, et l’autre dans laquelle on considère un zoom virtuel (dans le cadre de notre procédure en trois étapes). Pour obtenir des résultats comparables, et pour cette expérience uniquement, les deux approches sont évaluées sur les mêmes images d’entrées, avec la même résolution (1920×1080 ramené à un carré 1080×1080 par un rognage centré) et avec les mêmes discrétisations (2 mm au lieu de 5 mm utilisé dans l’estimation grossière). Concrètement, nous avons évalué les performances avec les images où le préhenseur est positionné approximativement au dessus du bloc, ce qui est une condition nécessaire pour l’estimation précise mais seulement un cas particulier pour l’estimation grossière. Les résultats sont reportés dans la Table 6.5. On peut voir que la procédure en trois étapes améliore énormément les performances. Ce qui fait la différence est que pour l’estimation de pose grossière, l’image doit être redimensionnée à 256×256 (soit un facteur de 4.2 sur chaque dimension de l’image) pour pouvoir la fournir au réseau de neurone de la procédure, alors que l’estimation de pose précise redimensionne un rognage centré sur le préhenseur dont la taille initiale est 432×432 (soit un facteur 1.7). Le rapport entre les résolutions des deux procédures est de 2.5 et se traduit donc naturellement par un rapport similaire pour la précision, et une différence considérable pour le taux de succès des erreurs sous 5 mm.

TABLE 6.5 – Taux de succès de l’estimation de pose avec une procédure en une seule étape par rapport à une procédure en trois étapes, sur la base de données ‘lab’.

Procédure	Une seule étape	Trois étapes	
Configuration	1 caméra 1 orientation du préhenseur	1 caméra 1 orientation du préhenseur	3 caméras 2 orientations du préhenseur
% ($e_x \leq 5$ mm)	20.5	61.1	90.8
% ($e_y \leq 5$ mm)	23.1	65.7	87.6
% ($e_\theta \leq 2^\circ$)	37.6	80.9	96.9
% ($e_x, e_y \leq 5$ mm, $e_\theta \leq 2^\circ$)	2.8	36.9	79.0

6.3.7 Comparaison avec d’autres méthodes

Nous avons éprouvé certaines difficultés à comparer notre méthode avec d’autres approches. Nous avons essayé diverses méthodes existantes pour détecter des blocs parallélépipédiques, telles que celle de Xiao et al. (2012), mais leurs performances sur les images de notre base de données d’évaluation n’étaient pas assez bonnes pour être utilisables. Alternativement, la détection de coins n’était pas assez fiable non plus pour localiser les blocs dans les images. En outre, la détection du bloc n’est qu’une étape du problème car, avec une approche par positionnement absolu, outre la calibration intrinsèque et extrinsèque de la caméra, le robot doit aussi être positionné dans le référentiel de la caméra pour pouvoir estimer la position relative du bloc. Pour construire un méthode de référence simple pour la comparer avec la nôtre, nous avons eu recours à des marqueurs visuels. Pour cela, nous avons construit une base de données supplémentaire, en tout point identique à notre base de données ‘lab’, à ceci près qu’un marqueur visuel a été apposé sur la face supérieure du bloc, comme illustré dans la Fig. 6.8. Avec cette base de données, nous pouvons détecter la position du bloc en utilisant le système de marqueurs ArUco (Garrido-Jurado et al., 2014, 2016) car la détection 2D des quatre coins du marqueur fonctionne bien : le marqueur est correctement détecté dans 95.0% des photos. Puisque, pour la base de données d’évaluation entière, nous utilisons le même bloc, avec une hauteur connue, et qu’il est toujours posé à plat sur la table plane, nous pouvons utiliser plusieurs positions connues du bloc, placé avec précision sur la table par le robot, pour calibrer indépendamment chaque caméra. Plus précisément, nous avons utilisé ces positions pour déterminer les coefficients d’une matrice d’homographie reliant le plan support au plan image. Comme le robot lui-même est utilisé pour placer le bloc sur la table à diverses positions, nous connaissons aussi la position relative du robot par rapport au bloc. Après cette étape de calibration, une détection 2D du bloc dans l’image se traduit directement en une pose sur la table plane, et ainsi en une pose relative par rapport à la base du robot. La Table 6.6 compare notre méthode aux résultats obtenus dans l’estimation de pose avec l’approche basée sur les marqueurs. (Les valeurs concernant notre méthode diffèrent de celles de la Table 6.2 parce qu’elles ne prennent pas en

compte seulement l'estimation précise mais la procédure en trois étapes complète, avec les possibles échecs lors de l'estimation de pose grossière et la détection du préhenseur.) Comme on peut s'y attendre, la méthode avec les marqueurs est plus performante que notre méthode, en ne considérant qu'une seule vue. Cependant notre méthode atteint des performances comparables lors de l'agrégation des vues des trois caméras et deux orientations du préhenseur ; et pourtant sans toutes les contraintes d'ordre pratique de l'approche avec les marqueurs.

Cette expérience souligne le fait que notre approche ne résout pas une tâche complètement nouvelle, ni une tâche qui ne pourrait pas être résolue avec les outils existants. Néanmoins, la performance d'une méthode "classique" dépend principalement de la qualité de la calibration de la caméra, et des algorithmes d'alignement 2D/3D du robot et du bloc. Or la calibration est difficile, potentiellement longue et n'est pas pérenne sur un chantier. De plus, nous avons trouvé difficile de trouver un algorithme d'alignement et de détection 2D-3D robuste et précis. Bien que les marqueurs peuvent rendre l'alignement 2D-3D facile, ils ont aussi des fortes contraintes pratiques car ils peuvent facilement être salis, endommagés ou partiellement cachés dans des scénarios réels, ce qui les rendrait inefficaces, sans compter qu'ils nécessitent un placement très précis sur les objets.

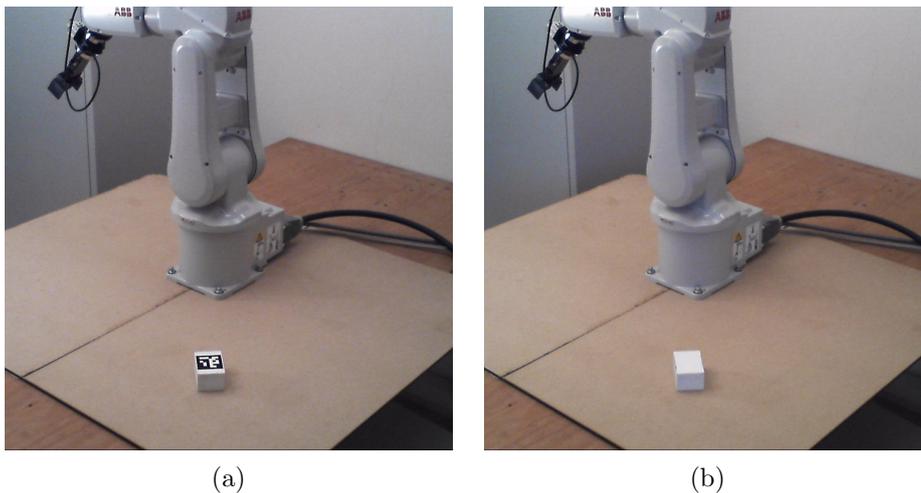


FIGURE 6.8 – Variante de la base de données 'lab' avec des poses identiques mais un marqueur visuel est apposé sur le dessus du bloc.

6.3.8 Sensibilité à l'environnement

Les résultats que nous venons de montrer par rapport à un environnement de laboratoire montrent le potentiel de notre méthode. Pour aller plus loin et évaluer la robustesse de notre approche, nous considérons des configurations plus difficiles, c'est-à-dire des conditions 'terrain' plus réalistes et celles plus défavorables 'défav'. Les résultats pour l'estimation de pose grossière sont visibles dans la Table 6.4, et pour l'estimation de pose précise dans les Tables 6.7 et 6.8. Comme attendu, les résultats sont meilleurs

TABLE 6.6 – Taux de succès de l’estimation de pose avec une méthode basée sur l’utilisation de marqueurs par rapport à notre méthode, sur la base de données ’lab’.

Méthode	Marqueurs	Notre méthode	
Configuration	1 caméra 1 orientation du préhenseur	1 caméra 1 orientation du préhenseur	3 caméras 2 orientations du préhenseur
% ($e_x \leq 5$ mm)	88.4	61.1	90.8
% ($e_y \leq 5$ mm)	95.0	65.7	87.6

dans l’environnement ’lab’. Les résultats dans un environnement plus réaliste ’terrain’ sont encore satisfaisant avec un taux de succès de 99.1% pour l’estimation grossière, et 70.2% pour l’estimation précise. Cependant, dans des conditions défavorables avec des morceaux de papiers que l’on peut facilement confondre avec le bloc (voir Fig. 6.6c), la chute de performance est plus grande. Le taux de succès pour l’estimation grossière où de nombreux distracteurs sont visibles est de 77.45%, et celui de l’estimation fine chute à 45.1%.

TABLE 6.7 – Taux de succès de l’estimation précise avec 3 caméras et 2 orientations du préhenseur sur les différentes bases de données réelles d’évaluation.

Base de données	’lab’	’terrain’	’défav’
% ($e_x \leq 5$ mm)	91.8	89.5	73.4
% ($e_y \leq 5$ mm)	88.5	86.8	71.0
% ($e_\theta \leq 2^\circ$)	97.9	93.0	86.5
% ($e_x, e_y \leq 5$ mm, $e_\theta \leq 2^\circ$)	79.9	70.2	45.1

TABLE 6.8 – Précision (moyenne et écart type) de l’estimation de pose précise avec 3 caméras et 2 orientations du préhenseur sur les différentes bases de données réelles d’évaluation.

Base de données	’lab’	’terrain’	’défav’
e_x (mm)	2.3 ± 1.8	2.3 ± 1.8	4.0 ± 4.4
e_y (mm)	2.6 ± 4.4	2.3 ± 2.0	4.0 ± 5.2
e_θ ($^\circ$)	0.7 ± 0.6	0.9 ± 0.7	1.1 ± 0.9

6.4 Conclusion

Nous avons introduit une nouvelle tâche d’estimation de pose relative sans calibration d’un objet par rapport à un bras robot. La tâche peut s’appuyer sur une unique

vue ou sur plusieurs vues, éventuellement avec différentes positions du bras, et avec des étapes intermédiaires possibles pour une estimation de pose plus précise. Nous avons aussi construit une base de données substantielle pour l'évaluation de cette tâche. Enfin, nous avons proposé une méthode générale pour réaliser cette tâche, ce qui fournit une méthode de référence. En effet, notre approche estime la pose du bloc par rapport au robot avec une précision moyenne – pour une configuration avec une seule caméra – de 6.2 mm à 8.9 mm en position et 1.1° à 1.3° en orientation, et une précision moyenne – pour une configuration à trois caméras – de 3.5 mm en position et 0.7° en orientation, et ces résultats, obtenus dans des conditions de laboratoires, se dégradent dans des configurations plus réalistes ou défavorables. Étant donnée la course limitée de l'ouverture de notre préhenseur, qui requiert une erreur de position inférieure à 5 mm et une erreur d'orientation inférieure à 2° , cela se traduit par un taux de succès global de 80% pour la saisie. Bien que ces résultats soient légèrement inférieurs aux méthodes se basant sur des caméras calibrées, ils montrent que l'estimation de pose relative peut être effectuée dans la pratique dans des configurations difficiles pour lesquelles la calibration de caméra est sensible ou ne peut pas être effectuée. Un des points forts de notre méthode est que l'apprentissage se fait seulement avec des images synthétiques, sans qu'aucune image réelle soit nécessaire, évitant ainsi la coûteuse et fastidieuse constitution d'une base de données d'entraînement. Les prolongations naturelles de notre approche incluent de traiter facilement différentes formes de robots, différentes tâches de saisie et des blocs de formes différentes, à différentes hauteurs et sur des surfaces non planes.

Chapitre 7

Conclusion et perspectives

7.1 Conclusion

Ce travail de thèse s’inscrit dans le contexte du développement de la robotique dans la construction. On s’intéresse ici à la construction robotisée de structures maçonnées complexes en ayant recours à de la vision artificielle. Nous avons abordé ce sujet via deux approches :

- par une conception des structures maçonnées qui permet d’obtenir des systèmes innovants en terme de comportement mécanique ou d’assemblage. La piste suivie ici est l’utilisation des assemblages topologiquement autobloquants pour pouvoir construire sans cintre et ainsi simplifier l’assemblage par des robots.
- par le développement d’algorithmes de vision artificielle adaptée aux contraintes d’un chantier. Ces algorithmes permettent de localiser et d’estimer la pose relative d’un bloc de maçonnerie par rapport à un bras robot.

L’état de l’art a d’abord été dressé dans une première partie.

Les assemblages topologiquement autobloquants ainsi que certaines de leurs propriétés ont d’abord été présentés dans le Chapitre 2. La manière de concevoir ces assemblages sur des surfaces courbes et leur utilisation dans le domaine de la construction ont été évoquées, ainsi que la coupe de joints particulière qui leur confère à la fois une certaine rigidité en flexion et un comportement ductile – même si le matériau est fragile.

Dans le Chapitre 3, nous avons présenté plusieurs approches pour l’étude de stabilité des maçonneries. En particulier, l’analyse limite (ou plus généralement le calcul à la rupture), le calcul par éléments finis continus et celui par éléments discrets. Nous avons conclu que, pour l’instant, l’analyse limite n’est pas appropriée pour l’analyse des maçonneries autobloquantes car la recherche de solutions d’équilibre est rendue difficile à cause de la stéréotomie particulière de ces maçonneries. De plus, elle surestime les chargements limites supportés car elle ne considère pas un critère de frottement de Coulomb non associé. Nous avons alors retenu le calcul par éléments discrets pour la suite du mémoire même si celui-ci nécessite des données supplémentaires (pas toujours connues

ou facilement accessibles) sur le comportement matériau, le comportement des joints, etc.

Enfin, nous avons présenté dans le Chapitre 4 certains algorithmes de vision artificielle en commençant d’abord par exposer brièvement l’apprentissage supervisé et notamment les réseaux de neurones. Puisque l’apprentissage se fait à partir de données synthétiques, nous avons également présenté différentes approches pour adapter l’apprentissage entre domaines différents. Enfin, nous avons présenté diverses techniques de localisation et d’estimation de pose.

Les contributions de cette thèse ont ensuite été exposées dans une seconde partie.

Une procédure de conception d’assemblages topologiquement autobloquants a d’abord été généralisée à la conception de structures courbes, à partir de maillages de quadrangles plans, dans le Chapitre 5. Les quadrangles de ces maillages n’étant pas forcément réguliers, nous avons montré qu’il faut vérifier certaines inégalités pour s’assurer du blocage topologique en translation. Ces inégalités nous ont permis de concevoir des assemblages plans dans lesquels les valeurs d’inclinaison des joints ne sont pas de signe alterné, contrairement à ce qui est trouvé dans la littérature. Nous avons également introduit la notion d’assemblage fortement topologiquement autobloquant. Celle-ci fait référence à l’existence d’assemblages dans lesquels chacun des blocs est bloqué en translation, mais un sous-ensemble – constitué de plusieurs de ces blocs – ne l’est pas. La généralisation de la procédure nous a permis de concevoir des structures maçonnées courbes dont la coupe des joints ne suit pas le principe de normalité. Nous avons conçu deux prototypes dont l’un contient des blocs en porte-à-faux et est constructible sans aucun cintre.

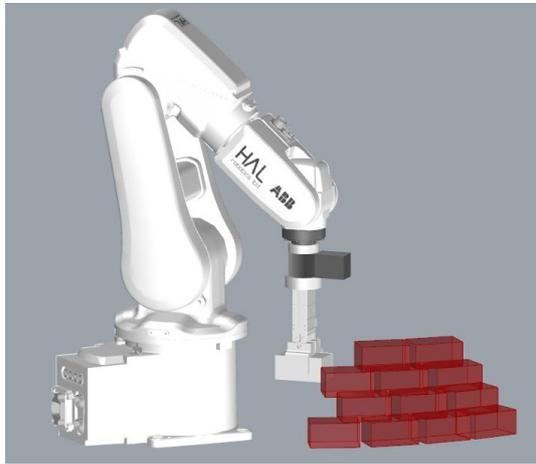
Nous avons ensuite (Chapitre 6) introduit une nouvelle tâche de vision artificielle, l’estimation de pose relative sans calibration d’un objet par rapport à un bras robot. Celle-ci peut s’appuyer sur une unique vue ou sur plusieurs vues, éventuellement avec différentes positions du bras, et avec des étapes intermédiaires possibles pour une estimation de pose plus précise.

Notre approche repose sur l’apprentissage d’un réseau de neurones convolutifs à partir uniquement de données synthétiques. Pour pallier à l’écart de domaine entre données réelles et données synthétiques, nous avons utilisé une technique de *domain randomization*, en générant de nombreuses images non réalistes mais d’une grande diversité, notamment par l’utilisation de nombreuses textures, poses du bloc et du robot, conditions d’éclairage, etc. Nous avons également construit une base de données substantielle, composée d’images réelles, pour l’évaluation de cette tâche.

Enfin, nous avons proposé une méthode générale pour réaliser cette tâche, ce qui fournit une méthode de référence. En effet, notre approche estime la pose du bloc par rapport au robot avec une précision moyenne – pour une configuration avec une seule caméra – de 6.2 mm à 8.9 mm en position et 1.1° à 1.3° en orientation, et une précision moyenne – pour une configuration à trois caméras – de 3.5 mm en position et 0.7° en orientation, et ces résultats, obtenus dans des conditions de laboratoires, se dégradent dans des configurations plus réalistes ou défavorables. Étant donnée la course limitée de l’ouverture de notre préhenseur, qui requiert une erreur de position inférieure à 5 mm et

une erreur d'orientation inférieure à 2° , cela se traduit par un taux de succès global de 80% pour la saisie. Bien que ces résultats soient légèrement inférieurs aux méthodes se basant sur des caméras calibrées, ils montrent que l'estimation de pose relative peut être effectuée dans la pratique dans des configurations difficiles pour lesquelles la calibration de caméra est sensible ou ne peut pas être effectuée. Un des points forts de notre méthode est que l'apprentissage se fait seulement avec des images synthétiques, sans qu'aucune image réelle soit nécessaire, évitant ainsi la coûteuse et fastidieuse constitution d'une base de données d'entraînement.

Un démonstrateur réel intégrant un bras ABB IRB 120, des blocs parallélépipédiques et trois webcams a été mis en place pour démontrer la faisabilité de la méthode. Il nous a permis de valider notre approche et d'en voir les limites en construisant entre autre un mur courbe à l'aide de briques parallélépipédiques, comme illustré sur la figure Fig. 7.1.



(a) Conception numérique d'un mur légèrement courbe. Le parcours d'outil a été généré grâce au logiciel HAL Robotics (<https://hal-robotics.com/>).



(b) Mur construit en utilisant notre algorithme de vision artificielle pour la localisation et l'estimation de pose des blocs.

FIGURE 7.1 – Conception et fabrication robotisée d'un mur

7.2 Perspectives

Les inégalités de blocage introduites dans le Chapitre 4 ne concernent que le blocage en translation. Il serait intéressant d'essayer de trouver des conditions similaires pour le blocage en rotation et étudier si l'un implique l'autre.

Le blocage fortement topologique a été exposé seulement sommairement dans ce mémoire. Pour prouver qu'un assemblage est fortement topologiquement autobloquant, il faut réussir à prouver que n'importe quel sous-ensemble connexe de blocs est bloqué. Résoudre ce problème de combinatoire est une perspective envisageable.

Il serait également intéressant d'étudier le blocage ponctuel arête-arête et non seulement celui plan-plan.

Bien qu'un prototype constructible sans cintre ait été présenté, un modèle reste à développer pour une compréhension plus fine du comportement de ces typologies structurelles. Comprendre les raisons de cette stabilité et le cheminement des efforts est une première étape à franchir afin de pouvoir optimiser la valeur d'inclinaison des joints en fonction, entre autre, du porte-à-faux voulu et des efforts en jeu, pour construire sans cintre.

Les blocs composant le prototype de corolle ont quelques désavantages pour une construction robotisée. Premièrement, ils sont localement très fins et donc fragiles si construits en pierre. De plus, leur forme n'est pas optimisée pour une préhension par un robot. Nous avons développé et sommes en train d'étudier un nouveau type de bloc qui se base sur le principe de ce prototype ; ces blocs étant moins fragiles et plus facilement saisissables par un robot. Ils permettront a priori de construire un plafond sans utilisation de coffrage et de manière robotisée.

Concernant les aspects de vision artificielle, les prolongations naturelles de notre approche incluent de traiter facilement différentes formes de robots et de blocs (notamment les blocs topologiquement autobloquants), à différentes hauteurs et sur des surfaces non planes. Nous avons effectué une estimation de pose à trois paramètres (deux de positions et un d'orientation). Une perspective possible est de généraliser l'approche d'estimation relative à de l'estimation de pose à six paramètres. De plus, il serait intéressant de généraliser l'approche au cas multi-blocs afin d'être plus proche de ce qui se passerait sur un chantier réel.

Enfin, ces développements liés à l'apprentissage de robots s'inscrit désormais dans le projet de recherche de l'I-Site Future DiXite (Digital Construction Site) dont l'objectif est de fédérer les compétences autour de la construction numérique et du chantier du futur (<https://dixite.future-isite.fr/>).

Bibliographie

- 3DEC (2013). 3DEC 5.0 Theory and Background. Fourth edition.
- Akbarzadeh, M., Van Mele, T., and Block, P. (2015). 3d graphic statics : Geometric construction of global equilibrium. In Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium.
- Ashby, M. F. and Bréchet, Y. J. M. (2003). Designing hybrid materials. Acta Materialia, 51(19) :5801–5821.
- Aubry, M., Maturana, D., Efros, A. A., Russell, B. C., and Sivic, J. (2014). Seeing 3D chairs : exemplar part-based 2D-3D alignment using a large dataset of CAD models. In Conference on Computer Vision and Pattern Recognition (CVPR), pages 3762–3769. IEEE.
- Autruffe, A., Pelloux, F., Brugger, C., Duval, P., Bréchet, Y., and Fivel, M. (2007). Indentation behaviour of interlocked structures made of ice : Influence of the friction coefficient. Advanced Engineering Materials, 9(8) :664–666.
- Bagi, K. (2014). When Heyman’s Safe Theorem of rigid block systems fails : Non-Heymanian collapse modes of masonry structures. International Journal of Solids and Structures, 51(14) :2696–2705.
- Barbosa, F., Woetzel, J., Mischke, J., Ribeirinho, M., Sridhar, M., Parsons, M., Bertram, N., and Brown, S. (2017). Reinventing construction : A route to higher productivity. Executive summary, McKinsey Global Institute.
- Besdo, D. (1985). Inelastic behaviour of plane frictionless block-systems described as cosserat media. 37.
- Block, P. (2009). Thrust Network Analysis. PhD thesis.
- Block, P., Ciblac, T., and Ochsendorf, J. (2006). Real-time limit analysis of vaulted masonry buildings. Computers and Structures, 84(29-30) :1841–1852.
- Bock, T. and Langenberg, S. (2014). Changing building sites : Industrialisation and automation of the building process. Architectural Design, 84(3) :88–99.

- Bock, T. and Linner, T. (2015). Synchronization of Organization, Building Structure, and Manufacturing Technology by Robot-Oriented Design, page 156–230. Cambridge University Press.
- Brocato, M., Deleporte, W., Mondardini, L., and Tanguy, J.-E. (2014). A Proposal for a New Type of Prefabricated Stone Wall. International Journal of Space Structures, 29(2) :97–112.
- Brocato, M. and Mondardini, L. (2010). Geometric methods and computational mechanics for the design of stone domes based on Abeille’s bond. In Advances in Architectural Geometry 2010, volume 1, pages 149–162. Springer Vienna, Vienna.
- Brocato, M. and Mondardini, L. (2012). A new type of stone dome based on Abeille’s bond. International Journal of Solids and Structures, 49(13) :1786–1801.
- Brocato, M. and Mondardini, L. (2015). Parametric analysis of structures from flat vaults to reciprocal grids. International Journal of Solids and Structures, 54 :50–65.
- Brugger, C., Bréchet, Y., and Fivel, M. (2008). Experiments and Numerical Simulations of Interlocked Materials. Advanced Materials Research, 47-50 :125–128.
- Chalhoub, M. (2010). Massifs rocheux : homogénéisation et classification numériques. Presses des Mines.
- Charnes, A. and Greenberg, H. (1951). Plastic collapse and linear programming. Bull. Amer. Math. Soc.
- Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2016). Synthesizing training images for boosting human 3D pose estimation. In 4th International Conference on 3D Vision (3DV), pages 479–488. IEEE.
- Choisy, A. (1883). L’art de bâtir chez les Byzantins. Librairie de la Société anonyme de publications périodiques.
- Choy, C. B., Stark, M., Corbett-Davies, S., and Savarese, S. (2015). Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Collet, A., Martinez, M., and Srinivasa, S. S. (2011). The MOPED framework : Object recognition and pose estimation for manipulation. The International Journal of Robotics Research (IJRR), 30(10) :1284–1306.
- Collet, A. and Srinivasa, S. S. (2010). Efficient multi-view object recognition and full pose estimation. In International Conference on Robotics and Automation (ICRA), pages 2050–2055. IEEE.
- Crivellaro, A., Rad, M., Verdie, Y., Moo Yi, K., Fua, P., and Lepetit, V. (2015). A novel representation of parts for accurate 3d object detection and tracking in monocular images. In International Conference on Computer Vision (ICCV). IEEE.

- Culmann, C. (1875). Die graphische Statik.
- Cundall, P. A. (1971). A computer model for simulating progressive, large-scale movement in blocky rock system. In Proceedings of the International Symposium on Rock Mechanics.
- Cundall, P. A. (1988). Formulation of a three-dimensional distinct element model-Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. International Journal of Rock Mechanics and Mining Sciences and, 25(3) :107–116.
- Cundall, P. A. and Hart, R. D. (1992). Numerical Modelling of Discontinua. Engineering Computations, 9(2) :101–113.
- D’Acunto, P., Ohlbrock, P. O., Jasienski, J.-P., and Fivet, C. (2016). Vector-based 3d graphic statics (part i) : Evaluation of global equilibrium. Proceedings of the IASS Annual Symposium 2016 ”Spatial Structures in the 21st Century”.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In International Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 886–893. IEEE.
- Deuss, M., Panozzo, D., Whiting, E., Liu, Y., Block, P., Sorkine-Hornung, O., and Pauly, M. (2014). Assembling self-supporting structures. ACM Transactions on Graphics, 33(6) :1.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet : Learning optical flow with convolutional networks. In International Conference on Computer Vision (ICCV), pages 2758–2766. IEEE.
- Duballet, R. (2014). Optimisation de structures en béton fibré ultra performant. Master’s thesis, École des Ponts ParisTech.
- Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., and Efros, A. A. (2018). Investigating Human Priors for Playing Video Games.
- Dugué, M. (2013). Experiments and simulations of interlocked materials. PhD thesis.
- Dyskin, A., Estrin, Y., Kanel-Belov, A., and Pasternak, E. (2003a). Interlocking properties of buckyballs. Physics Letters A, 319(3-4) :373–378.
- Dyskin, A. V., Estrin, Y., Kanel-Belov, A., and Pasternak, E. (2003b). Topological interlocking of platonic solids : A way to new materials and structures. Philosophical Magazine Letters, 83(3) :197–203.
- Dyskin, A. V., Estrin, Y., Kanel-Belov, A. J., and Pasternak, E. (2001a). A new concept in design of materials and structures : Assemblies of interlocked tetrahedron-shaped elements. Scripta Materialia, 44(12) :2689–2694.

- Dyskin, A. V., Estrin, Y., Kanel-belov, A. J., and Pasternak, E. (2001b). Toughening by fragmentation - how topology helps. Advanced Engineering Materials, 3(11) :885–888.
- Dyskin, A. V., Estrin, Y., Pasternak, E., Khor, H. C., and Kanel-Belov, A. (2003c). Fracture resistant structures based on topological interlocking with non-planar contacts. Advanced Engineering Materials, 5(3) :116–119.
- Estrin, Y., Dyskin, A. V., Pasternak, E., Schaare, S., Stanchits, S., and Kanel-Belov, A. J. (2004). Negative stiffness of a layer with topologically interlocked elements. Scripta Materialia, 50(2) :291–294.
- Fallacara, G. (2006). Digital stereotomy and topological transformations : reasoning about shape building. pages 1075–1092.
- Fallacara, G. (2016). Architectural stone elements : Research, design and fabrication.
- Fantin, M. (2017). Étude des rapports entre stéréotomie et résistance des voûtes clavées. PhD thesis.
- Fantin, M. and Ciblac, T. (2016). Extension of thrust network analysis with joints consideration and new equilibrium states. International Journal of Space Structures, 31(2-4) :190–202.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 32(9) :1627–1645.
- Feng, C., Xiao, Y., Willette, A., Mcgee, W., and Kamat, V. R. (2014). Towards autonomous robotic in-situ assembly on unstructured construction sites using monocular vision. In International Symposium on Automation and Robotics in Construction and Mining (ISARC).
- Feng, Y., Siegmund, T., Habtour, E., and Riddick, J. (2015). Impact mechanics of topologically interlocked material assemblies. International Journal of Impact Engineering, 75 :140–149.
- Fidler, S., Dickinson, S., and Urtasun, R. (2012). 3d object detection and viewpoint estimation with a deformable 3D cuboid model. In Advances in Neural Information Processing Systems (NIPS), pages 611–619.
- Fivet, C. (2016). Projective transformations of structural equilibrium. International Journal of Space Structures, 31(2-4) :135–146.
- Fleury, F. (2009). Evaluation of the Perpendicular Flat Vault Inventor ’ s Intuitions through Large Scale Instrumented Testing. In Third International Congress on Construction History.

- Frézier, A. F. (1737). La théorie et la pratique de la coupe des pierres et des bois, pour la construction des voûtes et autres parties des bâtiments civils et militaires, ou Traité de stéréotomie à l'usage de l'architecture. Jombert, C. A.
- Fukushima, K. (1980). Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36 :193–202.
- Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, pages 1180–1189, Lille, France.
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47(6) :2280 – 2292.
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Medina-Carnicer, R. (2016). Generation of fiducial marker dictionaries using mixed integer linear programming. Pattern Recognition, 51 :481 – 491.
- Giordano, A., Mele, E., and De Luca, A. (2002). Modelling of historical masonry structures : Comparison of different approaches through a case study. Engineering Structures, 24(8) :1057–1069.
- Girdhar, R., Fouhey, D. F., Rodriguez, M., and Gupta, A. (2016). Learning a predictable and generative vector representation for objects. In European Conference on Computer Vision (ECCV).
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587. IEEE.
- Girshick, R. B. (2015). Fast R-CNN. In International Conference on Computer Vision (ICCV). IEEE.
- Glasner, D., Galun, M., Alpert, S., Basri, R., and Shakhnarovich, G. (2011). Viewpoint-aware object detection and pose estimation. In International Conference on Computer Vision (ICCV), pages 1275–1282. IEEE.
- Glickman, M. (1984). The G-Block System of Vertically Interlocking Paving. In Second International Conference on Concrete Block Paving, pages 1–4.
- Godio, M., Stefanou, I., and Sab, K. (2018). Effects of the dilatancy of joints and of the size of the building blocks on the mechanical behavior of masonry structures. Meccanica, 53(7) :1629–1643.
- Grabner, A., Roth, P. M., and Lepetit, V. (2018). 3d pose estimation and 3d model retrieval for objects in the wild. In International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

- Hart, R., Cundall, P. A., and Lemos, J. (1988). Formulation of a three-dimensional distinct element model-Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. International Journal of Rock Mechanics and Mining Sciences and, 25(3) :117–125.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. arXiv preprint arXiv :1703.06870.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. CoRR, abs/1512.03385.
- Hejrati, M. and Ramanan, D. (2012). Analyzing 3D objects in cluttered images. In Advances in Neural Information Processing Systems (NIPS), pages 593–601.
- Heyman, J. (1982). The Masonry Arch. Ellis Horwood Ltd.
- Heyman, J. (1995). The Stone Skeleton : Structural Engineering of Masonry Architecture. Cambridge University Press.
- Hodaň, T., Matas, J., and Obdržálek, Š. (2016). On evaluation of 6D object pose estimation. In European Conference on Computer Vision Workshops (ECCVw), pages 606–619. Springer.
- Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. The Journal of Physiology, 160(1) :106–154.
- Huerta, S. (2001). Mechanics of masonry vaults : The equilibrium approach. 3rd International Seminar in Historical Constructions, Guimarães, Portugal, (February) :47–70.
- Huttenlocher, D. P. and Ullman, S. (1990). Recognizing solid objects by alignment with an image. International Journal of Computer Vision (IJCV), 5(2) :195–212.
- Kanel-Belov, A. J., Dyskin, A. V., Estrin, Y., Pasternak, E., and Ivanov-Pogodaev, I. A. (2010). Interlocking of convex polyhedra : Towards a geometric theory of fragmented solids. Moscow Mathematical Journal, 10(2) :337–342.
- Kao, G. T. C., Körner, A., Sonntag, D., Nguyen, L., Menges, A., and Knippers, J. (2017). Assembly-aware design of masonry shell structures : a computational approach. In IASS Annual Symposium, volume 141.
- Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). Ssd-6d : Making rgb-based 3d detection and 6d pose estimation great again. In International Conference on Computer Vision (ICCV). IEEE.
- Khandelwal, S., Siegmund, T., Cipra, R., and Bolton, J. (2012). Transverse loading of cellular topologically interlocked materials. International Journal of Solids and Structures, 49(18) :2394 – 2403.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4) :541–551.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In Proceedings of the IEEE, pages 2278–2324.
- Lemos, J. V. (2007). Discrete element modeling of masonry structures. International Journal of Architectural Heritage, 1(2) :190–213.
- Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. Journal of Machine Learning Research (JMLR), 17(39) :1–40.
- Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research (ISER), 37(4-5) :421–436.
- Lim, J. J., Pirsiavash, H., and Torralba, A. (2013). Parsing ikea objects : Fine pose estimation. In International Conference on Computer Vision (ICCV), pages 2992–2999. IEEE.
- Livesley, R. (1978). Limit analysis of structures formed from rigid blocks. International Journal for Numerical Methods in Engineering, 12(July 1977) :1853–1871.
- Livesley, R. K. (1992). A computational model for the limit analysis of three-dimensional masonry structures.
- Lourenço, P. B. and Ramos, L. F. (2004). Characterization of Cyclic Behavior of Dry Masonry Joints. Journal of Structural Engineering, 130(5) :779–786.
- Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence, 31(3) :355–395.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee.
- Masiani, R., Rizzi, N., and Trovalusci, P. (1995). Masonry as structured continuum. Meccanica, 30(6) :673–683.
- Massa, F., Marlet, R., and Aubry, M. (2016a). Crafting a multi-task CNN for viewpoint estimation. In 27th British Machine Vision Conference (BMVC).

- Massa, F., Russell, B. C., and Aubry, M. (2016b). Deep exemplar 2D-3D detection by adapting from real to rendered views. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages 6024–6033. IEEE.
- Molotnikov, A., Estrin, Y., Dyskin, A., Pasternak, E., and Kanel-Belov, A. (2007). Percolation mechanism of failure of a planar assembly of interlocked osteomorphic elements. Engineering Fracture Mechanics, 74(8) :1222–1232.
- Moseley, h. (1843). The Mechanical Principles of Engineering and Architecture.
- Mottaghi, R., Xiang, Y., and Savarese, S. (2015). A coarse-to-fine model for 3d pose estimation and sub-category recognition. In Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Mundy, J. L. (2006). Object recognition in the geometric era : A retrospective. In Toward category-level object recognition, pages 3–28. Springer.
- Novotny, D., Larlus, D., and Vedaldi, A. (2017). Learning 3d object categories by looking around them. In International Conference on Computer Vision (ICCV). IEEE.
- O’Dwyer, D. (1999). Funicular analysis of masonry vaults. Computers and Structures, 73(1-5) :187–197.
- Papadopoulos, C. T., Basanou, M., Vardoulakis, I., Boulon, M., and Armand, G. (1998). Mechanical behaviour of Dionysos marble smooth joints under cyclic loading : II. Constitutive modeling. In Mechanics of Jointed and Faulted Rock.
- Pegon, P., Pinto, A. V., and Gérardin, M. (2001). Numerical modelling of stone-block monumental structures. Computers and Structures, 79(22-25) :2165–2181.
- Peng, X. and Saenko, K. (2017). Synthetic to real adaptation with deep generative correlation alignment networks. arXiv preprint arXiv :1701.05524.
- Peng, X., Sun, B., Ali, K., and Saenko, K. (2015). Learning deep object detectors from 3D models. In International Conference on Computer Vision (ICCV), pages 1278–1286. IEEE.
- Pepik, B., Benenson, R., Ritschel, T., and Schiele, B. (2015). What is holding back convnets for detection ? In 37th German Conference on Pattern Recognition (GCPR), number 9358 in LNCS, pages 517–528. Springer.
- Pepik, B., Stark, M., Gehler, P., and Schiele, B. (2012). Teaching 3D geometry to deformable part models. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages 3362–3369. IEEE.
- Pinto, L. and Gupta, A. (2016). Supersizing self-supervision : Learning to grasp from 50k tries and 700 robot hours. In International Conference on Robotics and Automation (ICRA), pages 3406–3413, Stockholm, Sweden. IEEE.

- Pluijm, v. d. R. (1999). Out-of-plane bending of masonry : behaviour and strength. PhD thesis.
- Rad, M. and Lepetit, V. (2017). Bb8 : A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In International Conference on Computer Vision (ICCV). IEEE.
- Rad, M., Oberweger, M., and Lepetit, V. (2018). Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Rankine, W. J. M. (1858). A Manual of Applied Mechanics.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, Advances in Neural Information Processing Systems 28, pages 91–99. Curran Associates, Inc.
- Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data : Ground truth from computer games. In European Conference on Computer Vision (ECCV), pages 102–118. Springer.
- Rippmann, M. (2016). Funicular Shell Design. PhD thesis.
- Roberts, L. G. (1963). Machine perception of three-dimensional solids. PhD thesis, Massachusetts Institute of Technology (MIT).
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The SYNTHIA dataset : A large collection of synthetic images for semantic segmentation of urban scenes. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages 3234–3243. IEEE.
- Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. Psychological Review, pages 65–386.
- Salençon, J. (1983). Calcul à la rupture et analyse limite. Presses de l'école nationale des Ponts et Chaussées.
- Salençon, J. (2002). De l'Elasto-plasticité au calcul à la rupture. Editions Ecole Polytechnique.
- Schrems, M. and Kotnik, T. (2013). On the extension of graphical statics into the 3rd dimension. In Proceedings of the Second International Conference on Structures and Architecture.
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In 32nd International Conference on Machine Learning (ICML), pages 1889–1897.

- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2014). Overfeat : Integrated recognition, localization and detection using convolutional networks. In International Conference on Learning Representations (ICLR).
- Serratrice, J. and Durville, J. (1997). Description des roches et des massifs rocheux. Technical Report 211, Bulletin des laboratoires des Ponts et Chaussées.
- Shafaei, A., Little, J. J., and Schmidt, M. (2016). Play and learn : Using video games to train computer vision models. In 27th British Machine Vision Conference (BMVC).
- Simulia (2014). ABAQUS/Standard User’s Manual, Version 6.14.
- Smars, P. (2000). Etudes sur la stabilité des arcs et voutes. PhD thesis.
- Su, H., Qi, C. R., Li, Y., and Guibas, L. J. (2015). Render for CNN : Viewpoint estimation in images using CNNs trained with rendered 3D model views. In International Conference on Computer Vision (ICCV), pages 2686–2694. IEEE.
- Sun, B. and Saenko, K. (2014). From virtual to reality : Fast adaptation of virtual object detectors to real domains. In 25th British Machine Vision Conference (BMVC).
- Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., and Freeman, W. T. (2018). Pix3d : Dataset and methods for single-image 3d shape modeling. In Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Sundermeyer, M., Marton, Z.-C., Durner, M., Brucker, M., and Triebel, R. (2018). Implicit 3d orientation learning for 6d object detection from rgb images. In The European Conference on Computer Vision (ECCV).
- Tessmann, O. (2012). Topological interlocking assemblies. pages 211–219.
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In 30th International Conference on Intelligent RObots and Systems (IROS). IEEE/RSJ.
- Tulsiani, S. and Malik, J. (2015). Viewpoints and keypoints. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages 1510–1519. IEEE.
- Vazquez, D., Lopez, A. M., Marin, J., Ponsa, D., and Geronimo, D. (2014). Virtual and real world adaptation for pedestrian detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 36(4) :797–809.
- Vella, I. M. and Kotnik, T. (2016). Geometric Versatility of Abeille Vault. 2 :391–397.
- Weizmann, M., Amir, O., and Grobman, Y. J. (2016). Topological interlocking in buildings : A case for the design and construction of floors. Automation in Construction.

- Whiting, E. (2011). Design Of Structurally-Sound Masonry Buildings Using 3D Static Analysis. PhD thesis.
- Wu, J., Xue, T., Lim, J. J., Tian, Y., Tenenbaum, J. B., Torralba, A., and Freeman, W. T. (2016). Single image 3D interpreter network. In European Conference on Computer Vision (ECCV), pages 365–382. Springer.
- Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., and Savarese, S. (2016). Objectnet3d : A large scale database for 3d object recognition. In European Conference Computer Vision (ECCV).
- Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal : A benchmark for 3d object detection in the wild. In Winter Conference on Applications of Computer Vision (WACV). IEEE.
- Xiao, J., Russell, B., and Torralba, A. (2012). Localizing 3D cuboids in single-view images. In Advances in Neural Information Processing Systems (NIPS), pages 746–754. Curran Associates, Inc.
- Yong, H. T. D. (2011). Utilisation of topologically-interlocking osteomorphic blocks for multi-purpose civil construction. PhD thesis.

Annexe A

Annexes

A.1 Base de données d'apprentissage composée d'images synthétiques

Comme introduit en Section 6.1, nous avons généré trois bases de données synthétiques pour entraîner trois réseaux, un pour chaque sous-tâche :

1. Le robot et le bloc dans des poses aléatoires pour l'estimation grossière,
2. le robot avec le préhenseur en position verticale pointant vers le bas, pour la détection 2D du préhenseur,
3. des zooms du préhenseur maintenu verticalement et le bloc aléatoirement placé à ses côtés, pour l'estimation précise.

Nous avons créé un modèle simple de pièce dans laquelle un bras robot est placé sur le sol, et un bloc parallélépipédique de dimensions aléatoires est placé aléatoirement à ses côtés. Le bras robot avec lequel nous avons effectué nos expériences est un IRB120 de l'entreprise ABB, pour lequel nous avons un modèle 3D. Nous avons aussi un modèle 3D pour le préhenseur. Cependant, nous n'avons pas modélisé les câbles accrochés au robot et au préhenseur (pour comparaison voir Fig. 6.1). Nous avons considéré des configurations similaires à ce qui peut être trouvé dans la base de données d'évaluation composée d'images réelles, bien qu'avec plus de variations pour améliorer la robustesse. Les paramètres aléatoires que nous avons utilisé pour la génération d'images synthétiques sont les suivants :

- La taille de la pièce dans laquelle se trouve le robot est de $20\text{ m} \times 20\text{ m}$, de telle sorte que les murs sont visibles sur certaines des images.
- L'orientation et la position de la base du robot (qui fait environ $20\text{ cm} \times 30\text{ cm}$) sont choisies aléatoirement. (La hauteur du robot est de l'ordre de 70 cm et la longueur du bras de l'ordre de 50 cm).
- Les valeurs d'angle des articulations du robots sont tirées aléatoirement parmi toutes les valeurs possibles, excepté pour la tâche de localisation 2D du préhen-

seur et la tâche d'estimation précise, dans lesquelles l'extrémité du bras est placé verticalement au dessus du sol.

- Chaque dimension du bloc parallélépipédique (hauteur, largeur, longueur) est tirée aléatoirement entre 2.5 cm et 10 cm.
- Le bloc est posé à plat sur le sol avec sa position et son orientation tirées aléatoirement dans une zone de positions atteignables par le robot pour la sous-tâche d'estimation grossière, ou bien dans une zone carrée de 12 cm de côté en dessous du préhenseur pour la sous-tâche d'estimation précise.
- Toutes les textures, pour le sol, le robot et le bloc sont tirées aléatoirement parmi 69 textures différentes.
- Le centre de la caméra est tiré aléatoirement entre 70 cm et 130 cm du sol dans un manchon cylindrique de rayon minimum 1 m et de rayon maximum 2.8 m, centré sur le robot, comme illustré dans la Fig A.1.
- Pour les images destinées à l'estimation grossière (vues larges), la cible qui vise la caméra est tirée aléatoirement dans un cylindre, centré autour de la base du robot, de rayon 30 cm et de hauteur 50 cm.
- Pour les images destinées à l'estimation précise (vues zoomées), on effectue un rognage d'une vue large autour du centre du préhenseur avec un léger décalage aléatoire.
- La caméra pivote autour de son axe principal (ligne entre le centre de la caméra et sa cible) avec un angle tiré aléatoirement entre -8 et +8 degrés.
- La distance focale de la caméra est tirée aléatoirement entre 45 mm et 75 mm pour un capteur équivalent de taille 24 mm × 24 mm.
- Les images synthétiques sont de tailles 256 × 256 pixels.

Les images ont été générées avec le moteur de jeux vidéos Unreal Engine 4. La base de données créée pour l'estimation grossière consiste en approximativement 420k images (des exemples sont visibles dans la Fig. 6.4a), celle pour la détection 2D du préhenseur consiste en approximativement 280k images (exemples visibles dans la Fig. 6.4b) et celle pour l'estimation précise consiste en approximativement 600k images (exemples visibles dans la Fig. 6.4c). Nous avons utilisé 90% de ces images synthétiques pour l'apprentissage et les 10% restant pour la validation lors de l'apprentissage du réseau.

A.2 Base de données d'évaluation composée d'images réelles

Comme expliqué en Section 6.3.1, notre base de données d'évaluation est divisée en trois parties, correspondant aux trois configurations illustrées sur la Fig. 6.6.

1. pour les données 'lab', le robot et le bloc sont sur la table avec aucun distracteur, ni aucune texture particulière,
2. pour les données 'terrain', la table est recouverte de terre, sable et graviers rendant ainsi la surface inégale,

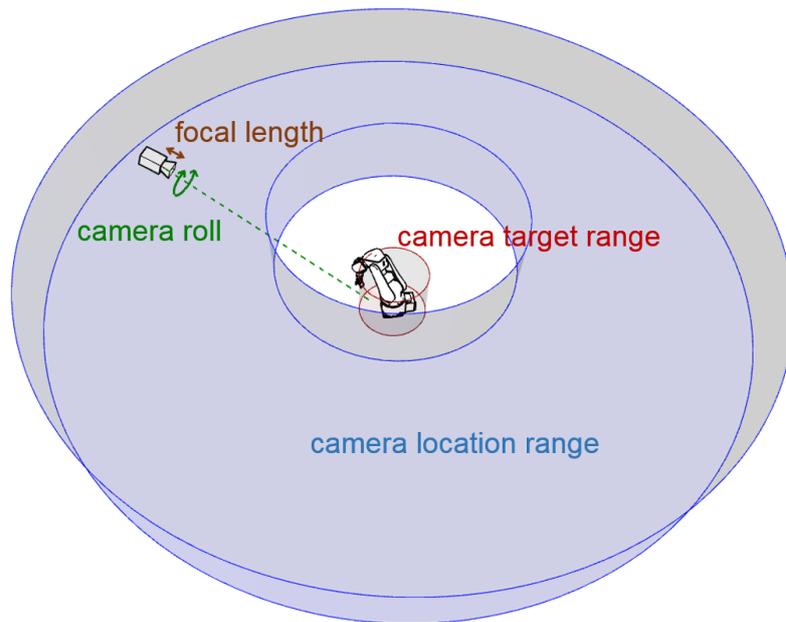


FIGURE A.1 – Représentation de certains paramètres définissant une configuration de pose relative entre le robot et une caméra.

3. pour les données 'défav', la table est couverte de morceaux de papiers qui peuvent être confondus avec les blocs parallélépipédiques.

Nous utilisons le robot lui-même pour déplacer précisément les blocs à des poses précises, ce qui fournit une mesure fiable de leurs positions et orientations relatives, pour chaque configuration. En pratique, le bloc peut légèrement se décaler de la position prévue au fur et à mesure que le robot le saisit, le déplace et le repose. Pour s'assurer qu'il n'y a pas de décalage, la position du bloc est vérifiée et réalignée toutes les dix poses. Dû à la course limitée des mors du préhenseur, nous n'avons considéré qu'un seul bloc de dimensions $5 \text{ cm} \times 8 \text{ cm} \times 4 \text{ cm}$. À noter cependant que notre méthode n'exploite pas cette information ; nous pensons qu'une méthode robuste est capable de traiter une large gamme de formes de blocs. Puisque nous voulons modéliser des situations dans lesquelles le robot peut saisir un bloc, nous avons restreint la portée du bras à la distance pour laquelle le préhenseur peut être maintenu vertical au-dessus du bloc, c'est-à-dire 0.505 m . Nous avons collecté les images de trois caméras à différents points de vue, regardant la scène légèrement par le dessus. Pour être sûr que le bloc soit visible dans la plupart des images, nous avons en fait considéré successivement différentes régions de la table d'expérience pour échantillonner les poses du bloc, déplaçant la caméra pour s'assurer d'une bonne visibilité pour chaque région. Les caméras sont déplacées manuellement sans soin particulier. La distance entre une caméra et le bloc est typiquement comprise entre 1 m et 2.5 m . L'angle maximum entre la caméra la plus à gauche et celle la plus à droite est de l'ordre de 120 degrés. Cette configuration est illustrée sur la Fig. 6.7. Pour chaque

position de bloc par rapport à la base du robot, nous avons considéré deux articulations principales du bras du robot : une configuration aléatoire du bras pour la sous-tâche de localisation grossière, et une configuration où le préhenseur est vertical, pointant vers le bas et positionné près du bloc pour la sous-tâche de localisation précise. Dans ce dernier cas, nous avons positionné le haut des mors du préhenseur 150 mm au dessus de la surface de la table, dans une position horizontale aléatoire dans un carré de 120 mm de côté autour du bloc. Nous avons en fait enregistré deux orientations du préhenseur en le pivotant autour de son axe vertical : une première orientation aléatoire puis une seconde orientation qui se déduit de la première par une rotation autour de l'axe vertical de 90 degrés (voir Fig. 6.5). Puisque l'orientation du préhenseur, par rapport à laquelle on définit l'estimation précise, peut être dure à estimer pour certaines configurations, utiliser deux orientations permet des prédictions plus précises. Au total, nous avons considéré approximativement 1300 poses (positions et orientations) du robot et du bloc. Ce qui mène à un total d'environ 12000 images, de taille 1080×1080 pour les vues larges et 432×432 pour les vues zoomées, avec les vérités terrains relatives pour la position et l'orientation. Les caméras utilisées sont les webcams eLight full HD 1080 de la marque Trust. Les paramètres intrinsèques de la caméra ne sont pas disponibles, ni estimés. Néanmoins, la distance focale a été déterminée approximativement comme valant 50 mm et dans les images synthétiques, la distance focale a été prise aléatoirement entre 45 mm et 75 mm. Les bases de données et modèles 3D sont disponible sur <http://imagine.enpc.fr/~loingvi/unloc/>.

A.2.1 Détails de l'architecture du réseau

Nous définissons ici ce que sont les cases pour les trois différents réseaux qui abordent chacun une sous-tâche. Le nombre de cases pour la dernière couche du réseau d'estimation grossière dépend de la taille de ces cases et de la portée maximale du bras robotique avec le préhenseur maintenu vertical, c'est-à-dire 0.505 m. En pratique, on définit des cases de 5 mm pour l'estimation grossière et de 2 mm pour l'estimation précise, ce que l'on peut visualiser par la grille fine rouge sur les figures Fig. 6.1a et Fig. 6.2. Ces figures donnent également une idée de la précision de la localisation par rapport aux dimensions du bloc et du robot. Pour l'estimation angulaire, nous avons utilisé des cases de 5 et 2 degrés respectivement. Pour le réseau de détection 2D du préhenseur, nous avons défini des cases dont la taille vaut 2% de la largeur de l'image. Puisque nous prédisons chaque dimension séparément, cela mène à 202 cases pour x et y et 36 cases pour θ pour le réseau d'estimation grossière, 60 cases pour x et y et 90 cases pour θ pour le réseau d'estimation précise, et enfin, 50 cases pour x et y pour le réseau de localisation du préhenseur.