

Mif11 - Ouverture à la recherche

Simulation Physique basé CGAL et GetFEM

Etudiants : VIOLENT LUCAS, KHEDIRI ABDESSAMED et TEKIN EMIR

Encadrant : Guillaume DAMIAND

I – Contexte

La simulation physique est un domaine clé de l'informatique graphique, permettant de reproduire le comportement d'objets tridimensionnels (3D), qu'ils soient rigides ou déformables, en se basant sur les lois de la mécanique pour une représentation réaliste dans le temps. Ce projet s'inscrit dans le cadre du développement de méthodes de simulation par éléments finis en utilisant deux bibliothèques C++ open source, GetFEM et CGAL.

La bibliothèque GetFEM fournit des outils pour la simulation physique, particulièrement adaptés pour la résolution de problèmes linéaires et non linéaires par éléments finis. CGAL propose une vaste gamme d'algorithmes de géométrie algorithmique ainsi que des structures de données permettant la modélisation d'objets complexes en 3D. Le package Linear Cell Complex de CGAL facilite la création et la manipulation d'assemblages de volumes qui serviront de support pour les simulations.

L'objectif principal de ce projet est d'intégrer les fonctionnalités de simulation physique de GetFEM avec les structures 3D représentées sous forme de "linear cell complex" dans CGAL, ouvrant ainsi la voie à une simulation physique détaillée et adaptable pour des objets modélisés dans CGAL. Ce travail requiert une bonne maîtrise du C++ avancé et implique un développement en programmation générique dans un environnement de code de haute qualité.

1. CGAL

La bibliothèque CGAL (*Computational Geometry Algorithms Library*) est une collection open-source d'algorithmes et de structures de données destinée à résoudre des problèmes géométriques complexes. Elle est largement utilisée dans des domaines comme la modélisation 3D, la robotique, la cartographie, et la simulation physique.

CGAL est structurée en plusieurs sous-packages, chacun conçu pour une problématique géométrique spécifique. Dans le cadre de ce projet, notre attention se porte sur le package *Linear Cell Complex* (LCC), développé par notre tuteur de projet. Ce package permet de modéliser des objets géométriques à plusieurs dimensions en utilisant des sous-objets appelés cellules (*cells*), chacun représentant une entité géométrique de dimension variable (points, arêtes, surfaces, volumes, etc.). Les connexions entre ces cellules forment un objet cohérent, permettant une description précise de la structure et de la topologie des objets complexes.

2. GetFEM

GetFEM est une bibliothèque open-source dédiée à la simulation numérique utilisant la méthode des éléments finis (FEM, *Finite Element Method*). Elle permet de modéliser et de résoudre des problèmes physiques dans des domaines comme la mécanique des structures, la thermique, et l'électromagnétisme. Ces simulations reposent sur des maillages géométriques qui représentent l'espace dans lequel les phénomènes physiques sont modélisés. GetFEM se distingue par sa capacité à manipuler des maillages et à appliquer des méthodes numériques avancées pour des simulations précises et performantes.

3. Le C++ et les Bibliothèques

Le C++ est un langage de programmation strictement typé, ce qui signifie que la conversion entre deux types de données doit être explicitement définie par le développeur. Les conversions implicites entre types, même s'ils sont similaires, ne sont pas automatiques. Cela pose parfois un défi lorsque l'on souhaite que différentes structures de données interagissent entre elles, comme c'est le cas dans l'intégration de CGAL et GetFEM.

Deux approches principales peuvent être envisagées pour gérer ce problème d'incompatibilité de types :

1. **Création d'une copie des données** : Cette méthode consiste à recréer un objet en initialisant toutes les données par copie. Cela garantit que les données sont adaptées à chaque bibliothèque, mais au prix d'un certain coût en performances.
2. **Transfert de la responsabilité des données** : Plutôt que de copier les données, cette méthode vise à transmettre la responsabilité des données à l'objet destinataire ou à utiliser un mécanisme de pointeurs pour que deux structures de données différentes puissent pointer vers un même ensemble de données, stocké en un unique exemplaire.

Ce projet est motivé par la recherche d'une intégration qui évite les coûts de copie en permettant une utilisation efficace et directe des données entre CGAL et GetFEM, en optimisant l'interopérabilité de ces deux bibliothèques dans des applications de simulation avancée.

II – Travail à réaliser

Ce projet vise à explorer l'intégration de deux bibliothèques spécialisées, CGAL et GetFEM, pour développer une application démontrant leur compatibilité et les bénéfices d'une interface optimisée. L'objectif est de prouver la faisabilité de cette intégration en minimisant les coûts de traitement et en garantissant une liaison efficace entre les deux bibliothèques. Le projet sera structuré en plusieurs étapes, allant de la prise en main des outils à l'évaluation des performances, en passant par la création d'un prototype. Ce programme, sera conçu en C++ (version 14), et destiné (dans la mesure du possible) à fonctionner sur plusieurs plateformes (Linux, Windows, MacOS) et pourrait inclure une partie graphique en fonction des avancements.

Étape 1 : Prise en Main des Bibliothèques

Avant toute tentative d'intégration, une familiarisation avec CGAL et GetFEM est indispensable. Cette étape consiste à :

- Installer et configurer chaque bibliothèque.
- Compiler et exécuter les exemples.
- Expérimenter leurs fonctionnalités de manière indépendante afin de maîtriser les structures de données
- Expérimenter plus en profondeur le module *Linear Cell Complex* de CGAL (module principale du projet).

Étape 2 : Développement du Prototype d'Intégration

L'objectif ici est de créer une preuve de concept pour démontrer une intégration optimale entre CGAL et GetFEM, sans recourir à des copies de données ou à des algorithmes coûteux :

- **Étudier l'interopérabilité des bibliothèques** : Identifier les points de compatibilité et d'incompatibilité entre CGAL et GetFEM.
- **Développer une interface d'intégration** : Concevoir une interface qui permettrait aux deux bibliothèques de fonctionner en harmonie en échangeant les données sans conversions complexes.
- **Implémenter un programme de démonstration** : Construire un programme simple pour valider cette intégration en lançant des simulations de maillage avec CGAL et GetFEM de manière cohérente.

Étape 3 : Tests et Validation

Après avoir établi une intégration de base, il s'agit de vérifier la solidité et la performance de cette approche :

- **Réaliser des simulations et comparer les résultats** : Utiliser les simulations fournies par CGAL et GetFEM pour valider la cohérence et la précision des résultats.
- **Mettre en place des méthodes de visualisation** : Afficher le maillage au cours de la simulation pour évaluer le comportement dynamique des objets.

Étape 4 : Finalisation et Perspectives

En fonction de l'avancement et du temps disponible, ces extensions pourront être explorées :

- **Exploration de méthodes avancées de simulation avec GetFEM** : Tester d'autres approches de simulation pour évaluer leur compatibilité avec CGAL.
- **Gestion des collisions** : Intégrer des mécanismes de détection et de résolution des collisions entre objets au sein des simulations.
- **Effectuer des benchmarks** : Comparer les performances de la solution développée avec celles d'une utilisation autonome de chaque bibliothèque, en particulier sur des aspects tels que la gestion des collisions et la rapidité des calculs.

III – Calendrier provisionnel

Temps estimé	Tâches
2 semaines	Étude des bibliothèques CGAL et GetFEM (installation, compilation d'exemples, tutoriels)
1 semaine	Conception de l'architecture du projet (multiplateforme, CMake, Git) et répartition des tâches
2 semaines	Création de l'interface entre les deux bibliothèques
2 semaines	Tests des premières simulations avec GetFEM
1 semaine	Rédaction des avancées dans le rapport
2 semaines	Développement de l'interface graphique
1 semaine	Rédaction finale du rapport et création de la vidéo de présentation