

Readme

1 Avancement

| Exercice | Achevé | Commentaire |
|-----------------|--------|--|
| P1 | ✓ | - |
| P3 | ✓ | - |
| P4 | ✓ | - |
| P5 | ✓ | La classe Masse a beaucoup évolué pendant le projet, pour s'adapter à nos exigences. Dans le repertoire de chaque exercice on peut trouver la version utilisée à ce moment là. |
| P6 | ✓ | - |
| P7 | ✓ | La classe Integrateur et ses sous-classes ont beaucoup évolué depuis cette première version, principalement pour l'absence de méthodes pratiques pour l'accroche des masses, utiles pour les tests proposés. Dans le repertoire de chaque exercice on peut trouver la version utilisée à ce moment là. Cette première version a été accompagnée par différents tests, utilisés pour vérifier le comportement du premier intégrateur et la réponse des masses et des ressorts. |
| P8 | ✓ | La première version de Tissu n'a pas beaucoup changé, mais elle a été au fur et à mesure enrichie par des méthodes utiles, tandis que d'autres ont été modifiées, par exemple la méthode <code>trou(size_t)</code> , qui est devenue la méthode <code>couper()</code> de la dernière version. |
| P9 | ✓ | Pour la classe Systeme on applique le même discours que pour Masse et Integrateur . |
| P10 | ✓ | - |
| P11 (P2) | ✓ | - |
| P12 | ✓ | - |
| P13 | ✓ | La classe Contrainte a été tout de suite codée en suivant les instructions fournies, puis, pour implémenter des extensions on a choisi de modifier sa conception et celles des classes filles. |
| P14 | ✓ | On a choisi d'implémenter et l'intégrateur de Newmark et l'intégrateur de Runge-Kutta, avec la possibilité de changer d'intégrateur pendant la simulation. On a prévu la possibilité d'afficher le nom de l'intégrateur utilisé grâce à un <code>QLabel</code> |

2 Partie graphique et heures de travail

La partie graphique de notre projet a été développée avec la bibliothèque graphique Qt5.

En moyenne on peut estimer d'avoir travaillé chacun quatre heures par semaine. Le travail a été reparté équitablement dans le binôme.

3 Rendu

On a choisi de soumettre le projet dans sa totalité, donc avec les répertoires de chaque exercice, afin de montrer l'avancement du travail et de la conception. On va maintenant expliquer le fonctionnement de notre projet, en explicitant le contenu de chaque étape et la façon de la compiler :

- **Exercices P1 → P4**

Le répertoire omonyme contient une version primordiale de la classe `Vecteur3D`, encore sans compilation séparée, dans le fichier `Vecteur3Dnonmod.cc`. Ensuite on a inclu les fichiers `Vecteur3D.cc` et `Vecteur3D.h`, qui compilent grâce au Makefile inclu pour donner l'exécutable `testVecteur`, qui satisfait les exercices P1, P3, P4.

- **Exercice P5**

Ce répertoire contient la première version complète de la classe `Masse` et son fichier `testMasse.cc`, qui compile les tests demandés.

- **Exercice P6**

L'exercice P6 contient la classe `Ressort` et son test, nommé `testRessort`, toujours à compiler avec le Makefile de ce répertoire.

- **Exercice P7**

Le répertoire contient les premières versions des classes d'intégration. De plus il contient les trois tests demandés, nommés `testIntegrateur i` , $i = 1, \dots, 3$, et plusieurs autres tests voués à vérifier le comportement de l'intégration et des forces comme les frottements. Le tout compile avec le Makefile fourni. On accompagne le tout par des graphiques réalisés avec `gnuplot`.

- **Exercice P8**

Dans ce répertoire on peut trouver les deux tests requis pour les Tissus (`testTissu1` et `testTissu2`), adaptés pour utiliser les intégrateurs. Les deux sont compilés par le même makefile.

- **Exercices P9 - P10**

Tout se passe comme auparavant. À noter que cet exercice utilise encore des versions désormais obsolètes des classes principales, qui ont évolué dans le projet final. L'exercice P9 crée un système simple, tandis que le P10 permet l'affichage texte d'une situation physique grâce à la classe `TextViewer`.

- **Exercice P11**

L'exercice P11 introduit les premiers pas de la partie graphique. Le répertoire

Qt_GL contient la simulation d'un pendule, le répertoire Qt_GL2 contient un tissu très simple et le Qt_GL3 contient la version graphique de l'exercice P10. De plus, le répertoire text contient la version texte du troisième exemple.

— **Exercices P12 → P14**

Ce répertoire contient les pas finals du projet, rassemblés dans les simulations de l'exercice 14. En particulier, on vérifie le comportement des nouveaux intégrateurs sur un tissu rectangulaire (sous-répertoire Qt_GL) et sur un pendule (Qt_GL2). On n'insère pas ici des tests spécifiques pour tous les types de contraintes et tissus, qui seront abondamment utilisés dans les extensions. Le répertoire **général** contient des plots relatifs aux nouveaux intégrateurs.

4 Projet final, extensions et How To

Le répertoire `final_and_extended_Project` accueille la version finale du projet et les simulations les plus intéressantes. Il y a deux sous-répertoires : le premier, `exemple1` contient des simulations réalistiques d'interaction entre des tissus et des corps modélisés grâce aux contraintes. Le deuxième, `exemples2`, contient des cas particuliers, où le tissu n'est pas soumis à la force de gravité, mais est sujet seulement aux forces appliquées par les contraintes. Tout sous-répertoire compile depuis le file `.pro` de `exemple1/2` avec la compilation `qmake-make`.

NB : Les versions soumises ont beaucoup de masses et le calcul pourrait être assez coûteux. Si jamais on conseille de diminuer les dimensions des tissus d'après les files `main_qt_gl`. En général on a apprécié une meilleure stabilité de l'intégrateur d'Euler, qui est conseillé dans ces simulations. Du coup on a appliqué des bornes au pas de temps pour pouvoir gérer la stabilité des autres intégrateurs.

Dans chaque simulation une fenêtre s'ouvrira, affichant le nom de l'intégrateur utilisé pendant la simulation. La touche *I* permet de changer d'intégrateur. En fermant le programme, il faut d'abord fermer la fenêtre d'intégration en appuyant `Backspace`, puis on peut fermer normalement la fenêtre graphique.

Encore, on a la possibilité de changer de méthode de visualisation en appuyant la touche *P*, pour visualiser masse et ressorts, seulement ressorts ou seulement masses. On déconseille de changer de méthode de visualisation pendant les calculs (risque d'instabilité), mais seulement pendant une pause de la simulation.

Plus en détail, `exemple1` :

— **Qt_GL0**

Répertoire de référence pour la correction : les fichiers `.h` dans **general** sont commentés, ainsi que le `main`, qui peut créer des exemples des différents tissus et contraintes. Le `main` donné construit un système vide. Il suffit simplement de décommenter les parties qu'on veut utiliser et changer les paramètres comme on désire pour ajouter les objets à la simulation.

— **Qt_GL1**

Simulation du secouement d'un tissu par une contrainte à impulsion sinusoïdale. Le tissu est fixé grâce à deux crochets, un desquels a une vitesse constante vers le haut.

— **Qt_GL2**

Simulation d'un tissu composé sur lequel tombent deux corps sphériques.

- **Qt_GL3**
Simulation d'un objet sphérique se déplaçant à vitesse constante qui rencontre deux rideaux, modélisés par des tissus rectangulaires disposés en vertical.
- **Qt_GL4**
Simulation d'une jupe composée de deux nappes circulaires trouées.
- **Qt_GL5**
Simulation d'un tissu tombant sur trois objets sphériques.
- **Qt_GL6**
Simulation de deux corps sphériques tombant sur un mouchoir aux bords accrochés.

exemple2 :

- **Qt_GL1**
Simulation simplifiée d'un mouchoir idéalement flottant sur un fluide visqueux, sur lequel tombent 3 corps sphériques.
- **Qt_GL2**
Simulation visuelle d'un champ gravitationnel engendré par des masses (corps sphériques) posées sur l'espace-temps (tissu aux bords accrochés).