Conception du Projet

Projet : Simulateur pour exercices de physique

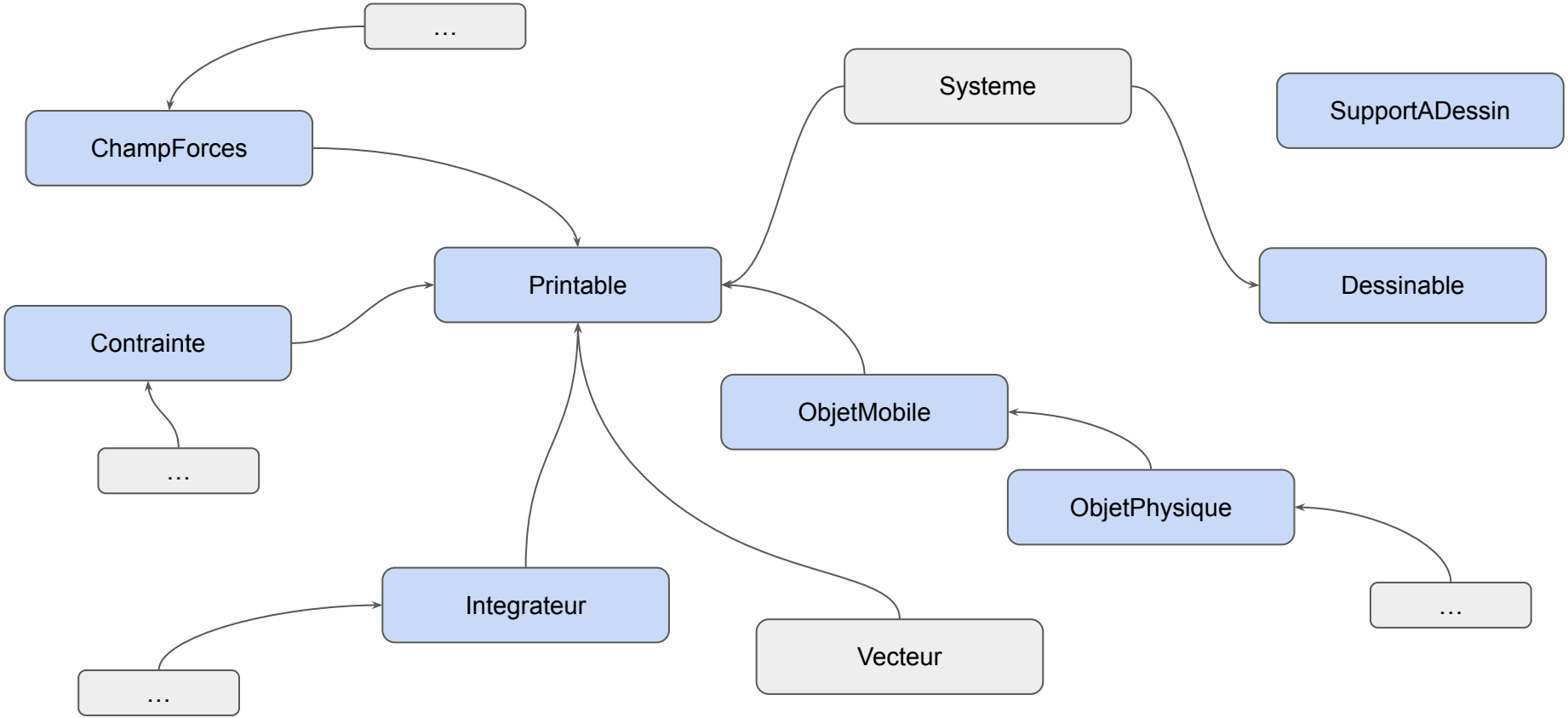
Matthias Calatroni
Valentino Lekimpe Calienes

Sommaire

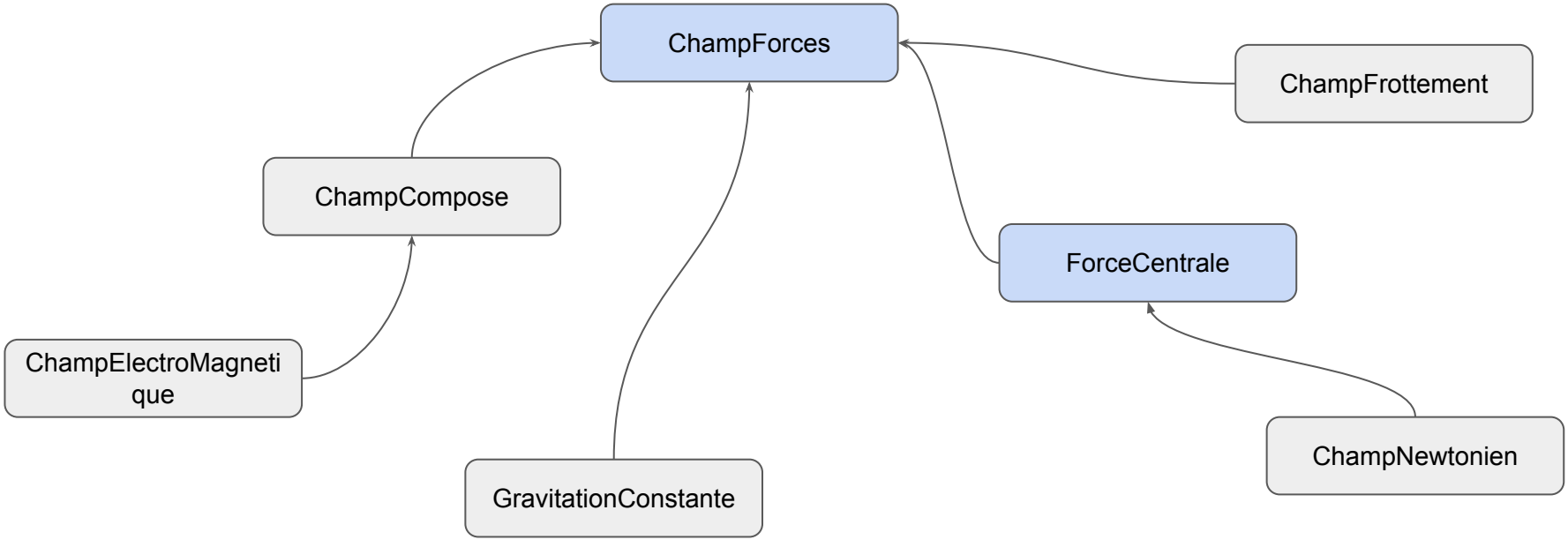
1. Diagramme des classes
2. Description des types des super classes ainsi que leurs sous classes.

Diagramme simplifié des classes du projet

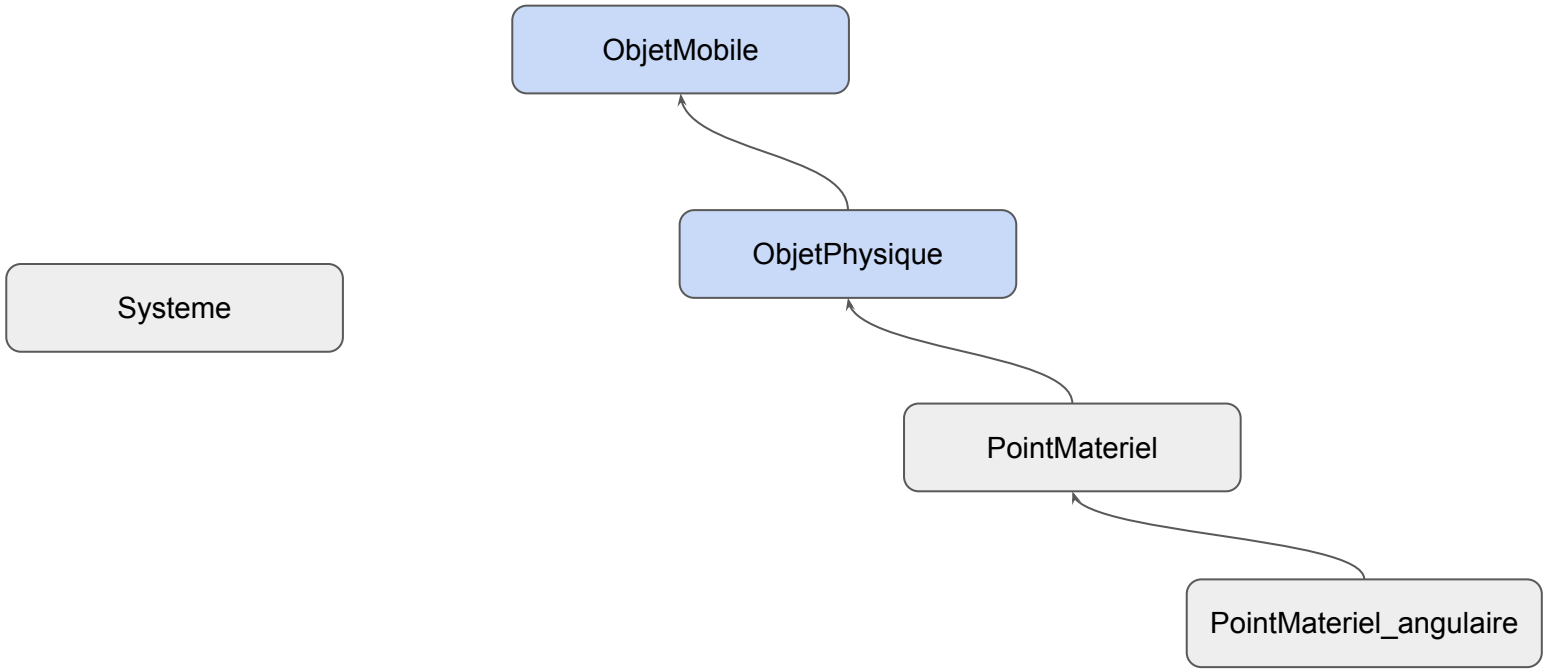
Classes Abstraites | Classes Instanciables



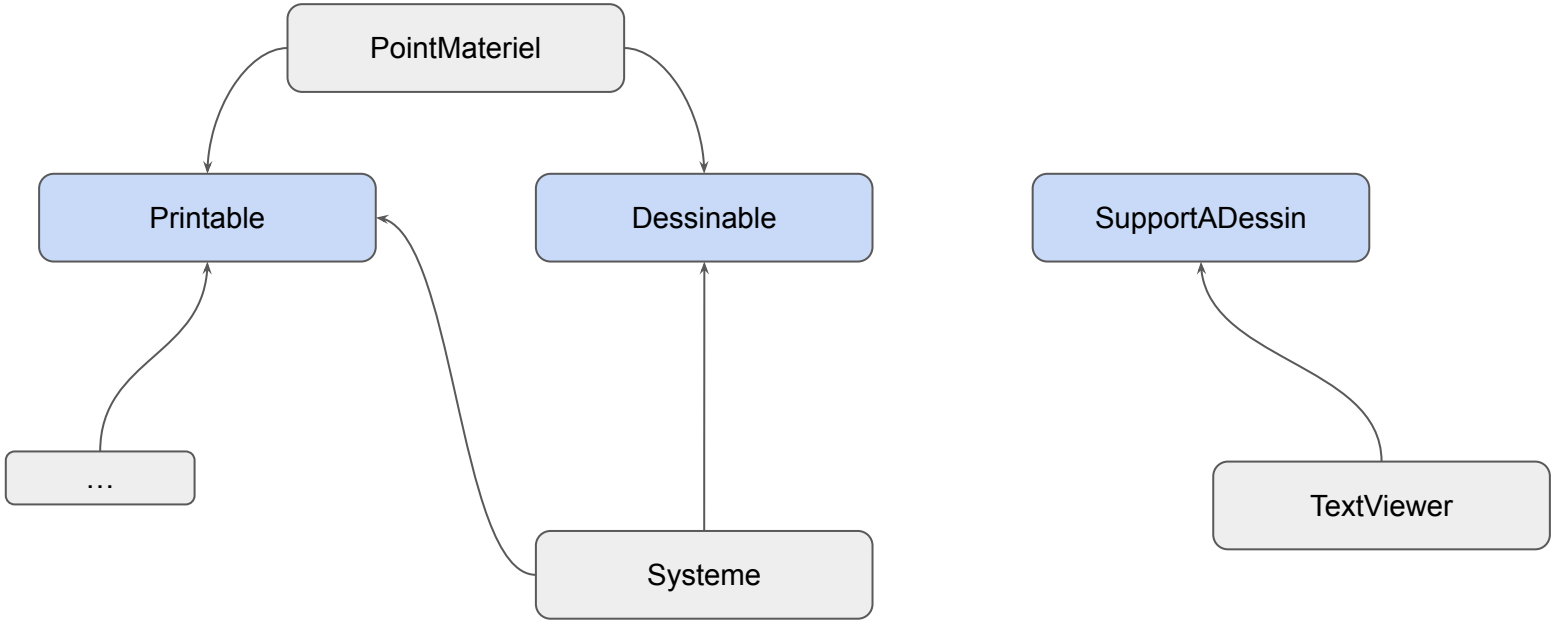
Implémentation des forces à travers des champs



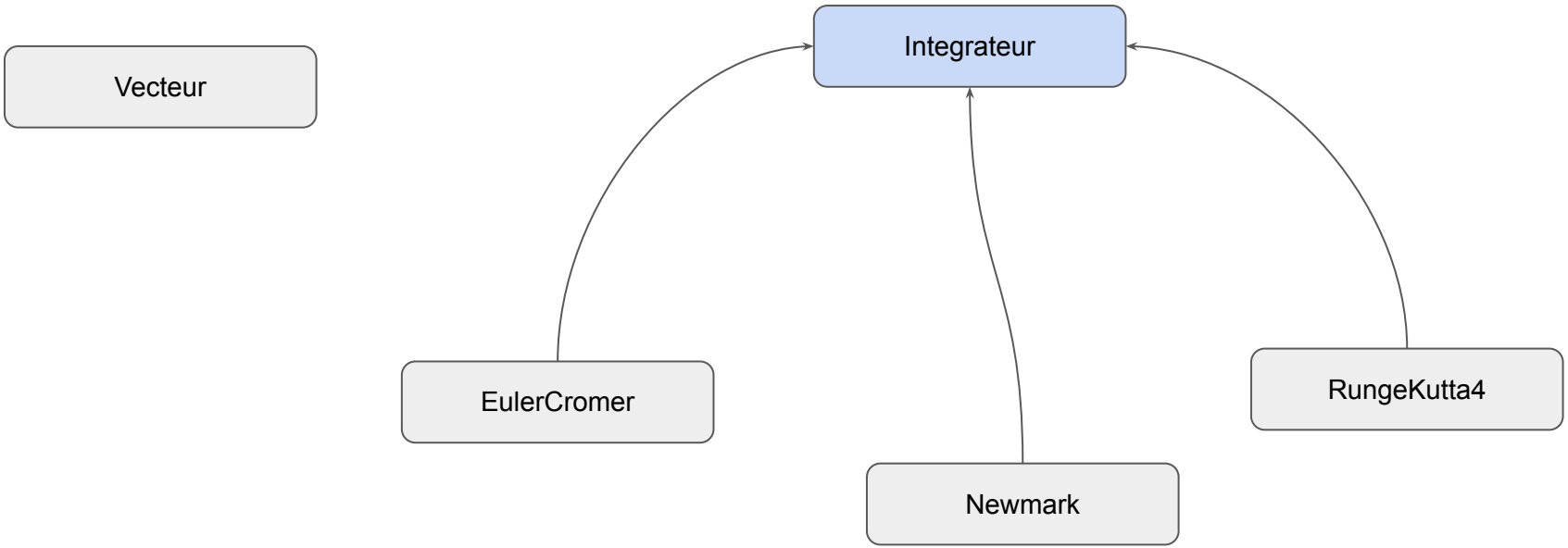
Implémentation de la simulation d’objets



Méthodes d’affichage



Outils de calculs et méthodes numériques



La classe Printable (abstraite)

Permet d’afficher tous les éléments de cette classe via la surcharge de l'opérateur << qui fait appel à la méthode abstraite affiche, possède aussi une méthode abstraite who am I pour dire expliciter la classe que l’on souhaite afficher

La classe Vecteur

représente des vecteurs de dimension quelconque

⚠️Pour toute opération entre 2 vecteurs, on prend la dimension du plus grand et on augmente la dimension du plus petit pour que les 2 soient égales

liste des constructeurs

- Constructeur par défaut par défaut
- Constructeur par 3 doubles (x, y, z)
- Constructeur par un int n pour initialiser un vecteur de 0.0 de dimension n
- Constructeur par un tableau de double
- Constructeur de copie par défaut

liste des opérateurs et méthodes

- ==
- !=
- -=
- -
- +=
- +
- * (multiplication par un scalaire)
- *=
- ^ (produit vectoriel)
- ~(normalisation)
- affiche
- whoami
- augmente
- get_coord
- set_coord
- get_dimension
- norme
- norme2

Les opérateurs existent aussi sous forme de méthode i.e pour + il y a la méthode addition, etc

Attributs :

- vecteur (vector de double)

La classe Système

Un système est une collection d'Objet Intégrables, de Champs de forces et de Contraintes à un instant t. Il permet, via la méthode evolution, de faire évoluer d'un temps dt, via l'intégrateur choisi, tous les objets Intégrables qu'il contient en fonction des Champs de Forces et des contraintes appliqué à chacun.

Il peut ensuite s'afficher sur un support à dessin par la méthode dessine_sur

liste des méthodes

- affiche
- whoami
- add_object
- add_constraint
- add_force_field
- change_integrator
- append_constraint (ajoute la contrainte i à l'objet j)
- append_force_field (ajoute le champ de force i à l'objet j)
- get_time
- dispaly_integrator
- get_obj (retourne l'objet i)
- get_champ (retourne le champs de force i)
- get_contr (retourne la contrainte i)
- evolve
- dessine_sur (dessine sur le support à dessin passé en paramètre)

Attributs :

- sys_objects (vector de ptr d'ObjetPhysique)
- sys_constraints (vector de ptr de Contrainte)
- sys_force_field (vector de ptr de ChampForce)
- integrator (ptr sur integrateur)
- time

Les Intégrateurs

Font évoluer des ObjetMobiles à partir de leur vecteur d'états en utilisant des méthodes numériques.

liste des méthodes

- evolue
- affiche
- whoami

Super Classe : Integrateur (abstraite)

Sous Classe 1 : EulerCromer

Sous Classe 2 : Runge-Kutta 4

Sous Classe 3 : Newmark

la méthode numérique utilisée est l'éponyme de chaque sous classe.

Affichage

Super Classe Abstraite Dessinable :
Type d'objets qui sont "dessinables" à partir d'une méthode dessine qui prend un SupportADessin en argument

Super Classe Abstraite SupportADessin :
Permet de définir des sous-classe agissant comme différentes méthodes d'affichage.

Sous Classe de SupportADessin : TextViewer
Affiche les simulations en mode texte en passant par le flot cout pour afficher sur le terminal.

Dessinable :

- méthode virtual dessine_sur(SupportADessin&), permet aux objets héritant de Dessinable de se dessiner sur le support à dessin passé par référence en faisant appel à la méthode dessine() du SupportADessin sur l'objet Dessinable courant.

SupportADessin :

- virtual dessine(objet) (méthode virutelle pure) redéfinie dans les sous-classes de SupportADessin comme Textviewer par exemple

TextViewer :
attribut :

- ostream& flot (flot stipulé à la construction, sur lequel l'objet est dessiné)

méthode :

- virtual dessine(objet) override→ prend l'objet et utilise sa surcharge de l'opérateur << pour le faire passer sur le flot. Pour afficher sur le terminal, on utilise le flot std::cout

Les Objets

Permettent la représentation des éléments pouvant évoluer dans l'espace

Super Classe : ObjetMobile (abstraite)

Sous Classe : ObjetPhysique (abstraite)

Sous Classe de ObjetPhysique : PointMateriel

Sous Classe de PointMateriel : PointMateriel_angulaire

liste des attributs (PointMateriel)
- masse
- charge
- dim_evo

liste des attributs (PointMateriel_angulaire)
- Lim

liste des méthodes (ObjetMobile)
- affiche
- whoami
- evolution (virtuelle pure)

liste des attributs (ObjetMobile)
- E (vecteur d'état)
- E_pr (dérivée du vecteur d'état)

liste des attributs (ObjetPhysique)
- charge

liste des méthodes (ObjetPhysique)
- force
- position
- vitesse
- add_contr
- add_champ

liste des méthode (PointMateriel)
- dessines_sur

liste des méthode (PointMateriel_angulaire)
- angle_max (limite les angle de Θ et φ

Ces classes sont aussi munie des setter et getter nécessaires

nous avons d'ailleur choisi de mettre la charge dans les attributs des ObjetPhysique au lieu de créer une nouvelle sous classe car nous utilisons du polymorphisme sur des ObjetPhysique dans notre classe Systeme

Les Champs de Forces

Permettent de représenter les différents types de forces à appliquer sur les objets

Super Classe : ChampForces (abstraite)

Sous Classe 1 : GravitationConstante

Sous Classe 2: ChampFrottement

Sous Classe 3 : ForceCentrale (abstraite)

Sous Classe de ForceCentrale : ChampNewtonien

Collection hétérogène de ChampForces : ChampCompose

Sous classe de ChampCompose : ChampElectromagnetique

liste des méthodes

- whoami
- force
- affiche
- quadratique_inverse (ForceCentrale)

Attributs :

- double g = 9.81 (GravitationConstante)
- double coeff_frottement (ChampFrottement)
- ObjetPhysique& centre (ForceCentrale)
- vector<ChampForces*> champs (ChampCompose)
- Vecteur champElec & champMag (ChampElectroMagnetique)
-

Les Contraintes

Contraignent le mouvement des objets à travers l'espace en modifiant la position et la vitesse des objets.

Super Classe : Contrainte (abstraite)

Sous Classe 1 : Libre

Sous Classe 2 : ContrainteSpherique

liste des méthodes

- affiche
- whoami
- applique_force (applique les forces subies par l'objet en fontion de la contrainte)
- position (retourne la position de l'objet)
- vitesse (retourne la vitesse de l'objet)

Attributs :

- rayon (que pour ContrainteSpherique) (distance à l'origine, la longueur de $\mathbf{u}_{\mathrm{O}}$)