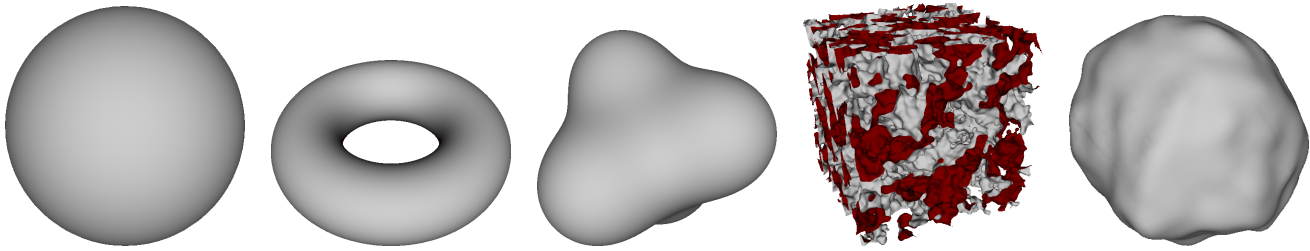


# TP 1 : Surfaces implicites



## 1 Introduction

L'objectif du TP est de manipuler les surfaces implicites. Pour rappel, une surface implicite est l'isopotentielle d'une fonction scalaire. Vous allez ici devoir coder de telles fonctions.

Vous pouvez effectuer ce TP seul ou en binômes. Un petit compte rendu est à rendre une semaine après la séance en salle machine.

## 2 Base de code

Pour la récupérer la base de code, allez à l'adresse suivante :

[https://team.inria.fr/imagine/files/2015/09/TP1\\_code.zip](https://team.inria.fr/imagine/files/2015/09/TP1_code.zip).

Pour l'utiliser, créez un dossier temporaire depuis lequel vous utiliserez CMake pour générer un Makefile avant de lancer la compilation :

```
cd TP1_code/build           # Acces au dossier de build
cmake ..                    # Creation du Makefile
make                         # Compilation + edition de liens
./marching_cube             # Execution
```

Ce programme gère la création d'une fenêtre GLFW contenant un contexte OpenGL (3.1). Ce contexte sert à l'affichage d'une scène 3D simple contenant un objet (initialement un cube), et permet de naviguer grâce aux contrôles suivants :

- ↑, ↓, ←, et → pour orienter la caméra
- PAGE UP et PAGE DOWN pour tourner la caméra
- Z et S pour zoomer/dézoomer respectivement
- Q, D, A et E pour se déplacer dans la scène
- SHIFT pour le mode lent

La classe `Mesh` contient un maillage, et se charge de la transmission des données d'affichage sur la carte graphique. Elle contient par ailleurs des méthodes statiques permettant de générer des primitives (cube et sphère), ainsi que des fonctions utiles comme le calcul automatique des normales, la suppression de points identiques, ou l'export vers un fichier OBJ. Enfin, elle contient la méthode `CreateIsoSurface()` qui implémente l'algorithme du *marching tetrahedra* que nous avons vu en cours. Cette dernière va vous être utile pour visualiser vos surfaces implicites. Elle prend en paramètres :

- le maillage dans lequel écrire

- la fonction définissant le potentiel scalaire dont on cherche à extraire l'iso-surface
- la valeur à laquelle on définit l'iso-surface
- une boîte englobante dans laquelle va être effectué l'algorithme
- une résolution à laquelle va être échantillonné l'espace de la boîte englobante

Notez que l'algorithme implémenté effectue une recherche de racine par dichotomie pour placer les sommets du maillage, ainsi qu'un calcul de normales par gradient de la fonction de potentiel.

### 3 Consignes

Imaginez différents types d'objets que vous souhaiteriez modéliser, et créez la fonction de potentielle permettant de les représenter.

Ces fonctions doivent être encapsulées dans la méthode `Eval()` d'une classe héritant de `ImplicitFunction`. Afin de visualiser l'isosurface de votre fonction, vous devez passer par la fonction `Mesh::CreateIsoSurface()` mentionnée précédemment, en lui fournissant les bons paramètres.

Commencez par des objets simples : sphères, cylindres, etc. Créez ensuite des classes permettant de combiner ces objets : mélange, interpolation, etc.

Pour toute les fonction que vous implémenterez, vous devez également coder le calcul de leur gradient dans la méthode `EvalDev()`. Celui-ci est utilisé lors du `marching tetrahedra` pour affecter les normales aux sommets. Dans un premier temps, vous pouvez passer par le calcul de normal offert par la fonction `Mesh::ComputeNormals()`, puis comparer les deux approches. Il est également possible d'obtenir le gradient par différences finies. Comparez les avantages et inconvénients du gradient analytique par rapport au schéma de différence finie.

Afin d'observer l'effet d'ajout de bruit dans votre fonction, une classe `PerlinNoise` vous est fournie, qui implémente un bruit de Perlin en 3D. À vous de créer la classe de fonction implicite l'utilisant et et combiner celle-ci avec d'autres primitives pour en observer les effets.

### 4 Rendu

Le rendu doit contenir les images des objets que vous avez généré, avec les équations que vous avez utilisé pour définir les champs correspondant. Expliquez également votre schéma de différence finie pour le calcul du gradient, et notez vos observations.

Mettez le tout dans un pdf s'appellant `MS_TP1_NOM1_NOM2.pdf` que vous m'enverrez par mail avec l'objet `[MS][TP1] Nom1 Nom2` en mettant en copie les deux membres du binôme.

### 5 Contact

Lors de la scéance, n'hésitez pas à poser vos questions, pour des problèmes théoriques ou de code. Vous pouvez ensuite poser vos questions par mail : [ulyse.vimont@inria.fr](mailto:ulyse.vimont@inria.fr). Le compte rendu est également à envoyer à cette adresse.