Chapitre 5: Matériaux

INF5071 — Infographie

Alexandre Blondin Massé

Université du Québec à Montréal

Hiver 2019

Plan

Lumière

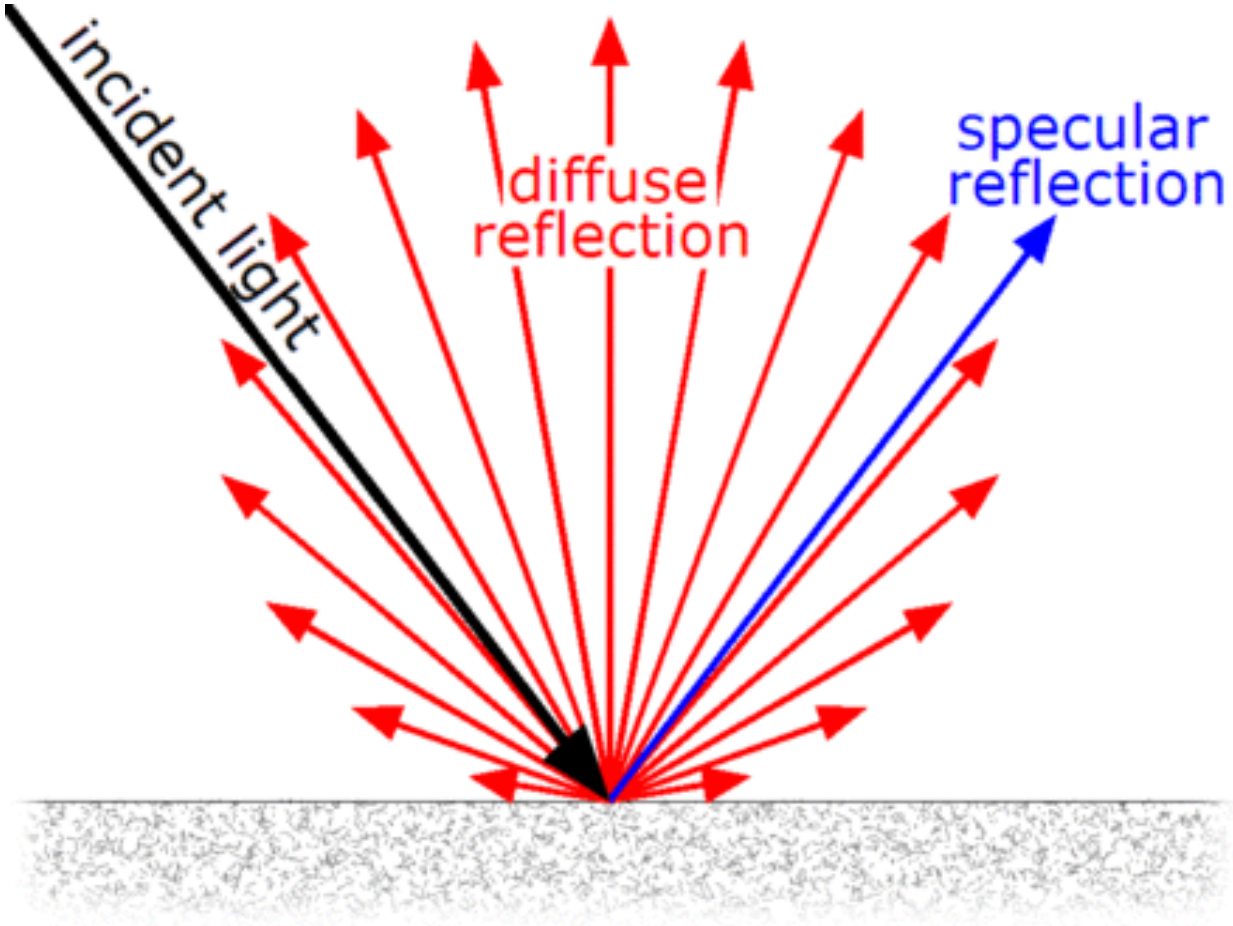
Couleur et lumière

- Un objet ne peut avoir une **couleur** que si une **source lumineuse** existe quelque part
- Autrement, il sera **complètement noir**
- Lorsqu'un **rayon de lumière** frappe un objet, celle-ci est **réfléchie** et obéit à différentes lois

Quelques phénomènes

- Réflexion **diffuse**: réflexion dans toutes les directions
- Réflexion **spéculaire**: qui tient compte de l'observateur
- **Diffraction**: changement de direction des rayons lumineux

Réflexions diffuse et spéculaire



(source: Wikipedia)

Modèle de Lambert

Un des modèles les plus simples et très utilisé en infographie

Soient

- $\vec{N}$ un vecteur **normal unitaire** à la surface
- $\vec{L}$ un vecteur **unitaire** en direction du point sur la surface vers la lumière
- i_L un **scalaire** indiquant l'intensité de la lumière

Alors l'intensité de **diffusion** est

$$i_D = i_L(\vec{N} \cdot \vec{L})$$

Réflexion spéculaire

- Une limite du modèle de Lambert est qu'il ne tient pas compte de l'**observateur**
- Peu importe l'angle, la texture apparaît de la même façon
- Or, la **position de l'observateur** devrait influencer le **rendu**
- Par exemple, si on se trouve

$\rightarrow$ **face** à une surface réfléchissante, on est **aveuglé**

$\rightarrow$ **de côté**, on peut **voir** correctement

- Pour tenir compte de cet aspect, plusieurs modèles ont été proposés: Phong, Blinn-Phong, Cook-Torrance, *Cel-shading*, ...

Développement de modèles

Mise en contexte

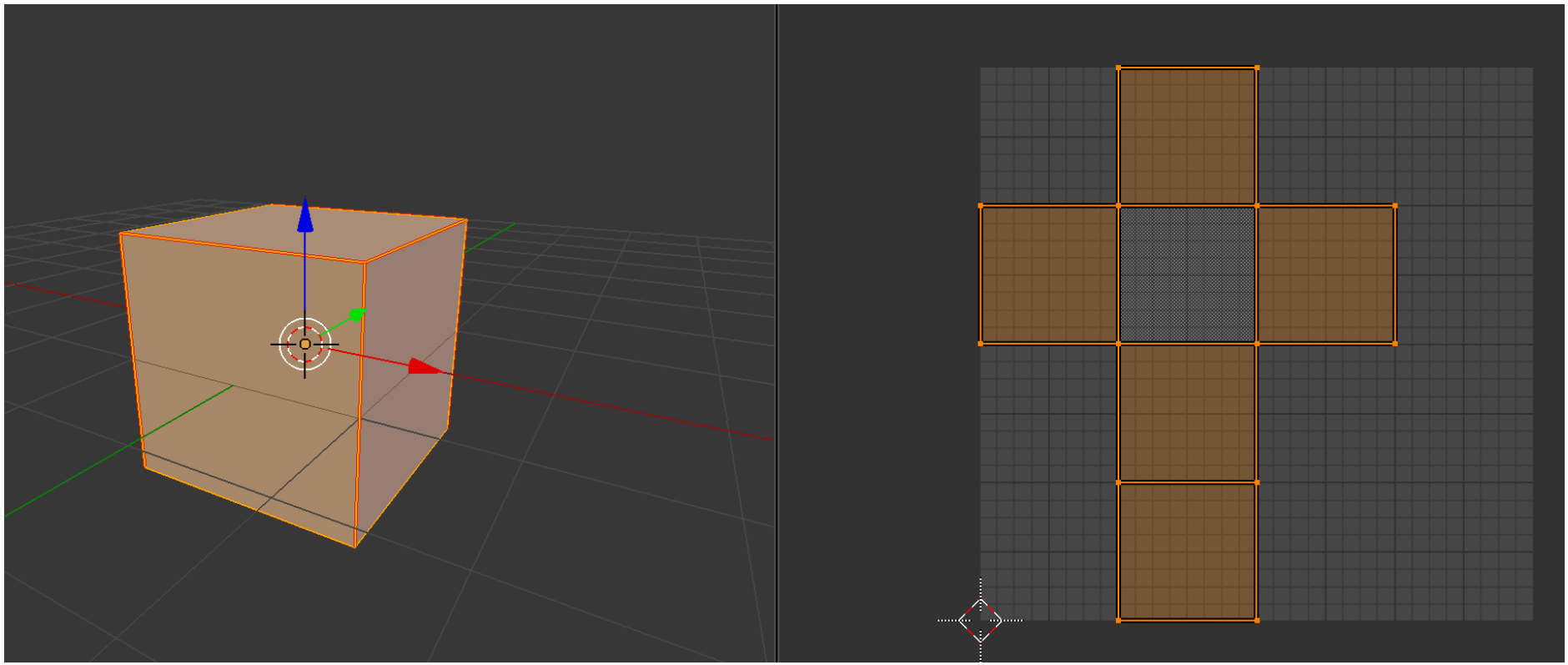
- Une **texture** est une image 2D
- **Problème**: les modèles sont des objets 3D
- On doit donc établir une **relation** entre les deux
- Essentiellement, il s'agit de construire une **fonction**

$$\begin{aligned} \text{unwrap} \quad : \quad \text{Modèle3D} \subseteq \mathbb{R}^3 \quad &\rightarrow \quad [0, h] \times [0, w] \\ (x, y, z) \quad &\rightarrow \quad (u(x, y, z), v(x, y, z)) \end{aligned}$$

- Autrement dit, on doit associer à chaque point 3D de notre modèle un point 2D dans un rectangle.

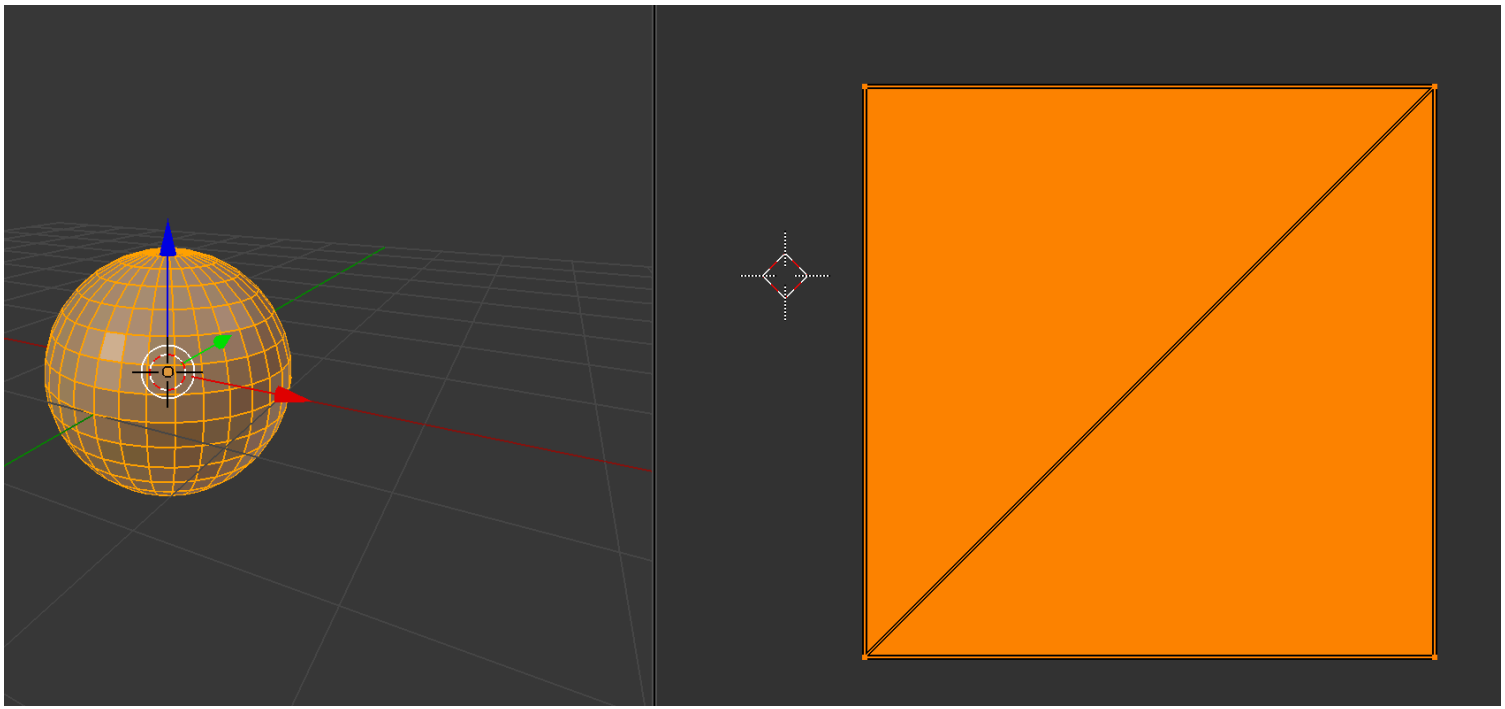
Développement ou projection

- Cette relation entre modèle et image est appelée **projection**
- Le nombre de **dimensions** de l'espace diminue
- On appelle ce processus **développement** du modèle (en anglais, *UV-unwrapping*)



Développement d'une sphère

- Le **développement** d'un modèle n'est pas toujours facile à déduire du modèle
- Par exemple, comment devrait-on développer une **sphère**?
- Même pour le cube de la diapositive précédente, si on développe, on obtient une projection **non utilisable**

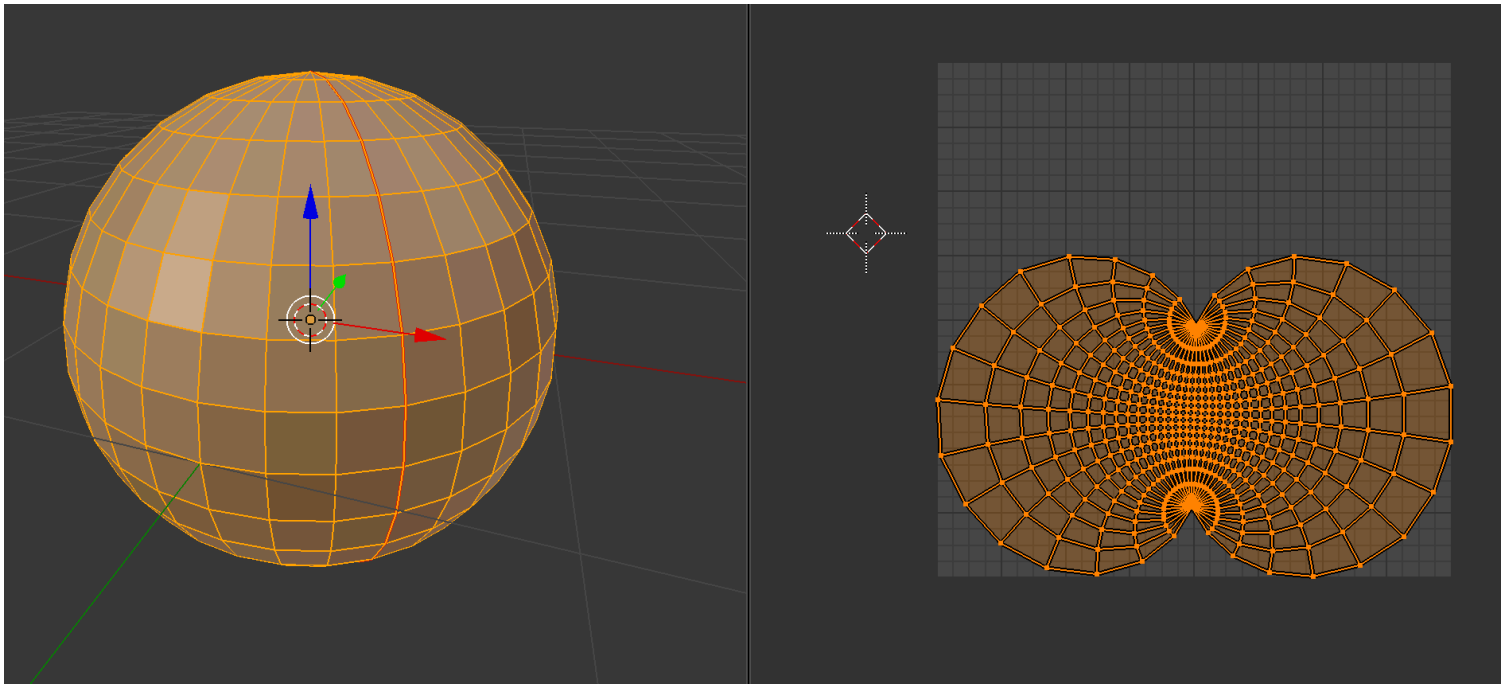


Couture (*seam*)

- La stratégie consiste à utiliser des **coutures** (en anglais, on dit *seam*)
- Une couture est simplement une arête le long de laquelle on accepte qu'il y ait une **discontinuité**
- Il faut donc choisir les **coutures** avec soin, si on veut s'assurer que la discontinuité n'est pas trop apparente
- Dans Blender, on identifie une couture en choisissant certaines arêtes et entrant la commande `CTRL-E`, suivi de *mark seam*
- Les coutures apparaissent alors en **rouge**

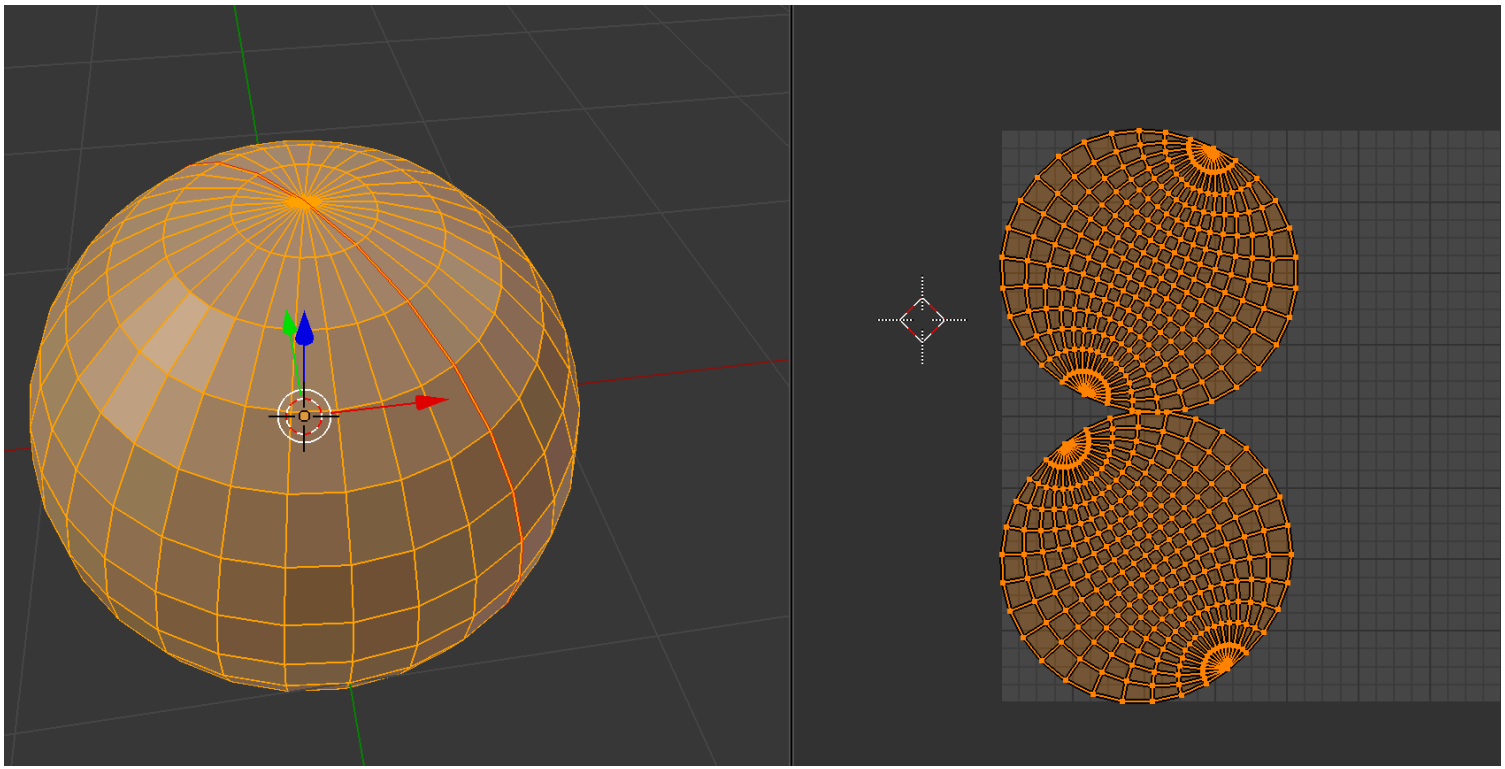
Exemple (1/4)

- Dans l'image ci-bas, il y a une **couture** le long d'un **méridien**



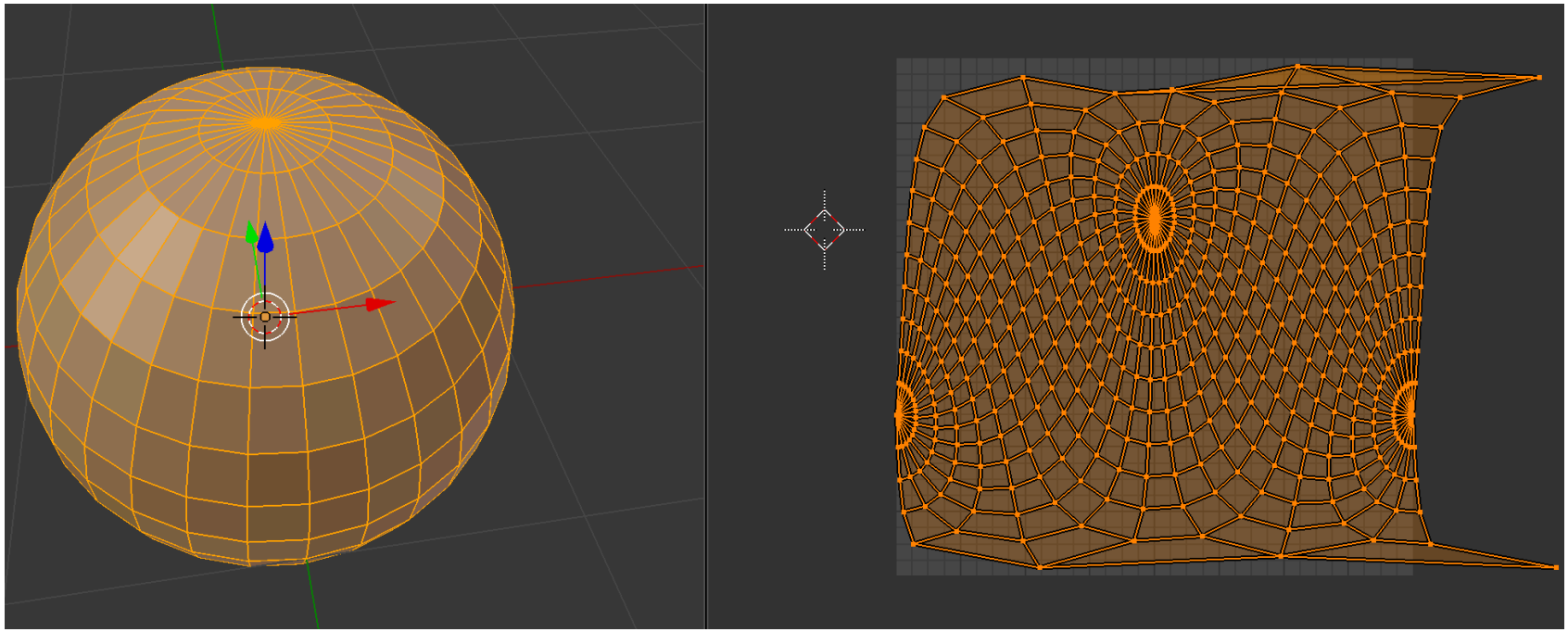
Exemple (2/4)

- Le développement est bien différent si on ajoute une couture faisant un **tour complet**



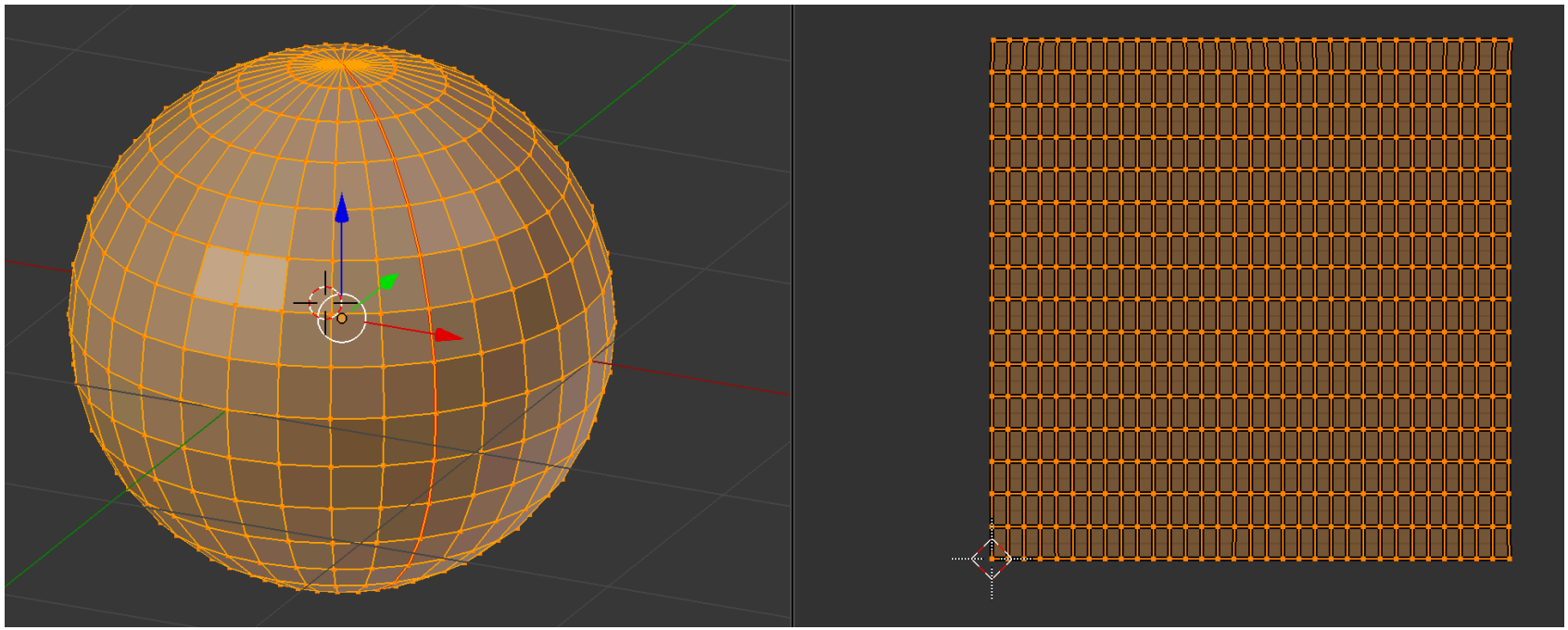
Exemple (3/4)

- On peut aussi faire une **projection sphérique**



Exemple (4/4)

- La projection sphérique devient **rectangulaire** si on remplace les pôles par plusieurs sommets



Choisir les coutures

Pour choisir les coutures, il faut une certaine pratique:

- Il faut minimiser les discontinuités trop **apparentes**
- C'est en particulier important pour les textures **périodiques**
- On sépare là où il y a un changement de **matériau**
- Ensuite, on prend une suite d'arêtes **moins visibles** où la coupure sera moins apparente

$\rightarrow$ **Exemple**: derrière l'objet

Texture

Texture

- Formellement, une **texture** est simplement une image 2D
- Nous allons aussi qu'on peut aussi créer des **textures 3D**
- Ces dernières sont simplement une **collection d'images** combinées d'une façon **spécifique**

$\rightarrow$ une texture de **relief** (*bump map*);

$\rightarrow$ une texture de **spéculaire**;

$\rightarrow$ selon les **vecteurs normaux**, etc.

Concevoir une texture

Logiciels

- Souvent, il y a un **aller-retour** entre des logiciels de modélisation 3D et les logiciels pour les images 2D
- De la même façon, certaines images **produites par Blender** sont ensuite modifiées à l'aide de logiciels 2D

Textures manuelles

- Il est aussi possible de créer des textures **manuellement**
- En anglais, on dit *model painting*

Génération procédurale

- On utilise des fonctions **mathématiques**
- **Exemples**: bruit de Perlin, diagramme de Voronoi, algorithme du carré et du losange, fonction sinusoïdales, ...

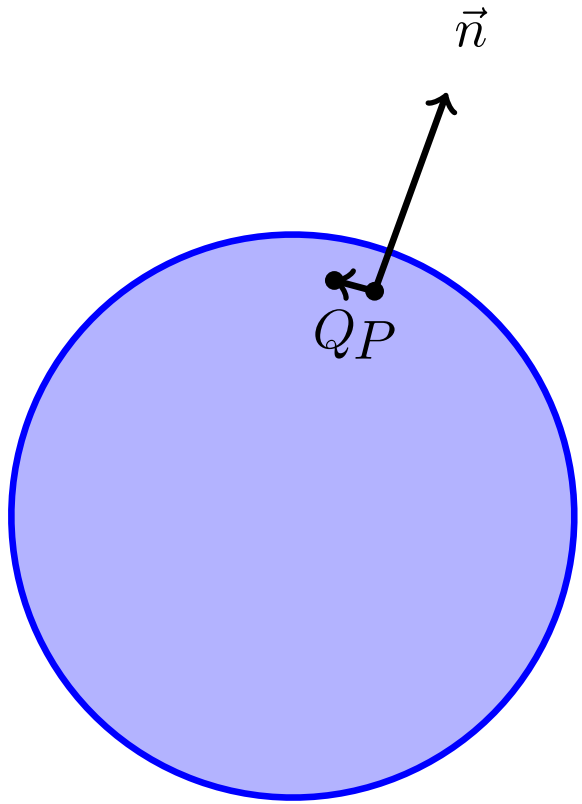
Dimensions d'une texture

- Souvent, on choisit les dimensions d'une texture pour qu'elles soient des **puissances** de 2;
- Cela permet d'avoir une meilleure qualité lors d'un **redimensionnement**
- *Mipmapping*: stratégie qui consiste à stocker des copies de plus en plus **plus petites** d'une image
- Ensuite, on utilise l'**interpolation** pour représenter les **dimensions intermédiaires**

Vecteur normal et espace tangent

Vecteur normal

- Soit S une surface et P un point de S
- Un **vecteur normal** à S en P est un vecteur perpendiculaire à tout vecteur non nul $\overrightarrow{PQ}$, où Q est un point **proche** de P, c'est-à-dire que $\|PQ\| \to 0$



Vecteur normal à un maillage

Question

Étant donné une surface, comment calcule-t-on un vecteur normal à cette surface en un point donné ?

Réponse

Ça dépend...

- Si constituée de faces **triangulaires**, il suffit de calculer un vecteur normal pour chaque face
- S'il s'agit d'une surface **mathématique**, il est possible de déduire une formule pour calculer un vecteur normal à partir de son **équation cartésienne** ou de sa **représentation paramétrique**

Surface implicite

Définition

Une **surface implicite** est définie comme l'ensemble des points (x, y, z) qui vérifient une équation de la forme

$$F(x, y, z) = 0,$$

où F est une fonction.

Exemples

- Si $F(x, y, z) = x^2 + y^2 + z^2 - r^2$, où r est une constante positive, alors l'équation décrit une **sphère** de rayon r centrée à l'origine.
- Si $F(x, y, z) = z^2 - x^2 - y^2$, alors l'équation décrit un **cône** évoluant dans la direction de l'axe x.

Gradient

Définition

Soit $f(x, y, z)$ une fonction réelle. Le **gradient** de f est défini par

$$\vec{\nabla} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right).$$

Remarque

Le gradient est un **vecteur**

Exemple

Si $f(x, y, z) = ax + bx \sin y - e^z$, alors

$$\vec{\nabla} f = (a + b \sin y, bx \cos y, -e^z).$$

Vecteur normal à une surface implicite

Théorème

Soit $F(x, y, z) = 0$ l'équation d'une **surface implicite** S et (x_0, y_0, z_0) un point situé sur la surface S. Alors le gradient

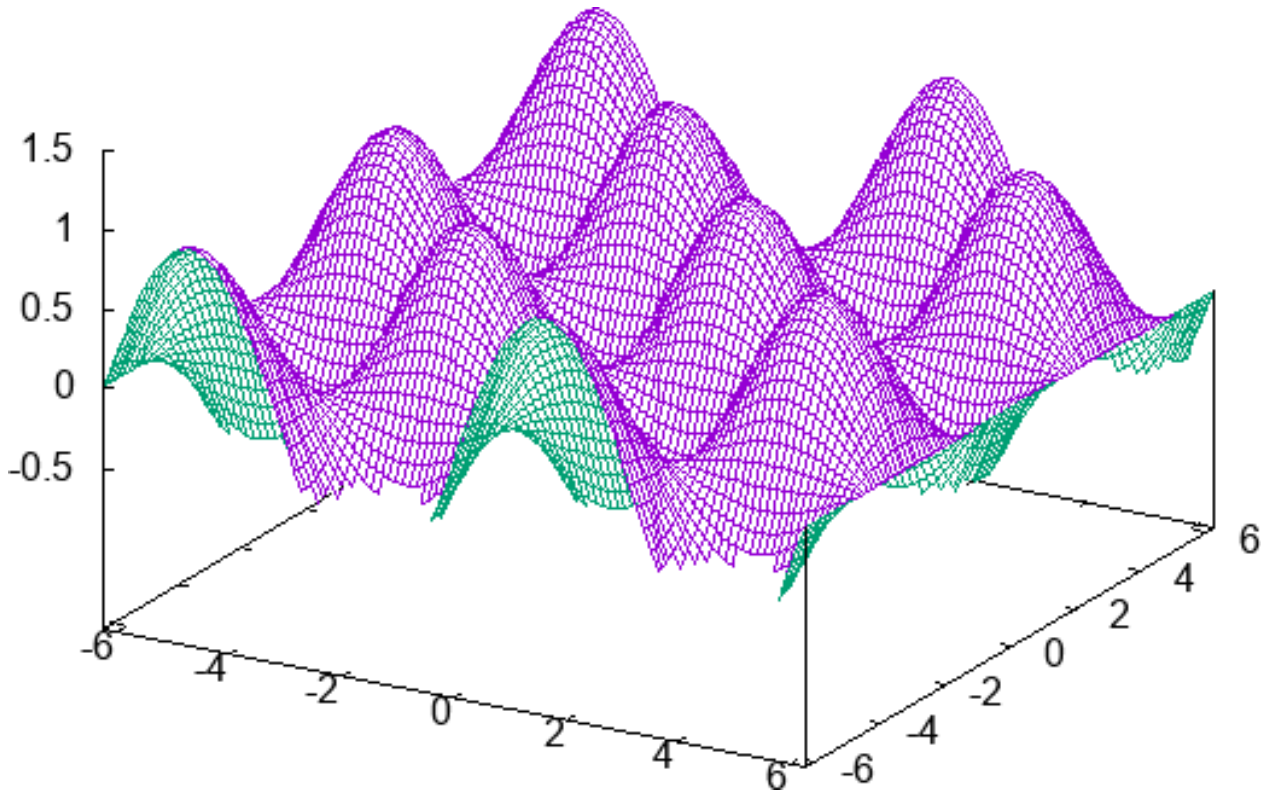
$$\overrightarrow{\nabla F}(x_0, y_0, z_0)$$

est **normal** à la surface S.

Questions

Calculez un vecteur normal **unitaire**

1. à la sphère centrée à l'origine de rayon $\sqrt{13}$ au point $(1, 2, 3)$
2. au cône d'équation $z^2 = x^2 + y^2$ au point $(0, 1, 1)$
3. à la surface donnée par l'équation $z = \sin(x)\cos(y)$ au point $(\pi/4, \pi/4, 1/2)$



Surface paramétrée

Rappel

Définition

Soit S une surface. On dit que la fonction vectorielle

$$\vec{s}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in D$$

est une **paramétrisation** de la surface S si

$$\text{Image}(\vec{s}) = S.$$

Vecteur normal d'une surface paramétrée

- Soit $\vec{s}(u, v)$ une paramétrisation d'une surface S
- Alors les vecteurs $\vec{s}_u(u, v)$ et $\vec{s}_v(u, v)$ sont **tangents** à la surface S au point (u, v)
- De plus, $\vec{s}_u(u, v)$ et $\vec{s}_v(u, v)$ sont **linéairement indépendants**
- Par conséquent, un **vecteur normal** à S au point (u, v) est donné par

$$\vec{n} = \frac{\vec{s}_u \times \vec{s}_v}{\|\vec{s}_u \times \vec{s}_v\|}.$$

Question

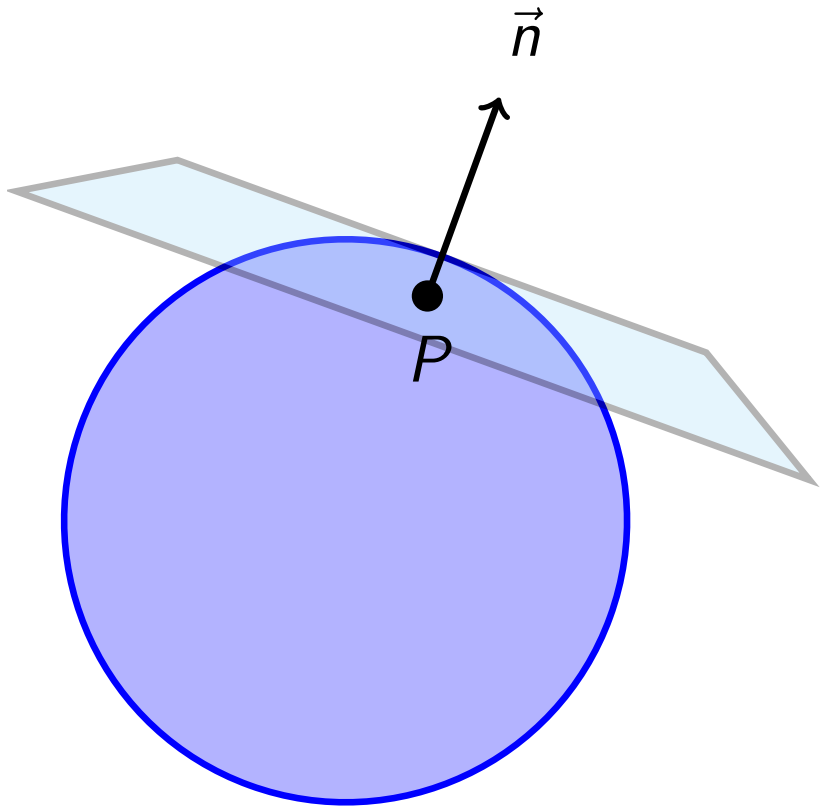
- Soit T le **tore** de grand rayon R et de petit rayon r
- Alors une paramétrisation possible de T est

$$\vec{s}(u, v) = ((R + r\cos u)\cos v, (R + r\cos u)\sin v, r\sin u)$$

1. Trouvez un point P qui se situe **sur** le tore T
2. Calculez ensuite un vecteur normal à T au point P

Espace tangent

- Soit S une surface, P un point de S et $\vec{n}$ un vecteur normal à S en P
- Le **plan tangent** à S en P est le plan passant par P ayant $\vec{n}$ comme vecteur normal



Cartes de normales

Champ de vecteurs

Définition

Un champ de vecteurs est une fonction de la forme

$$\vec{V} : D \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

en 2D et

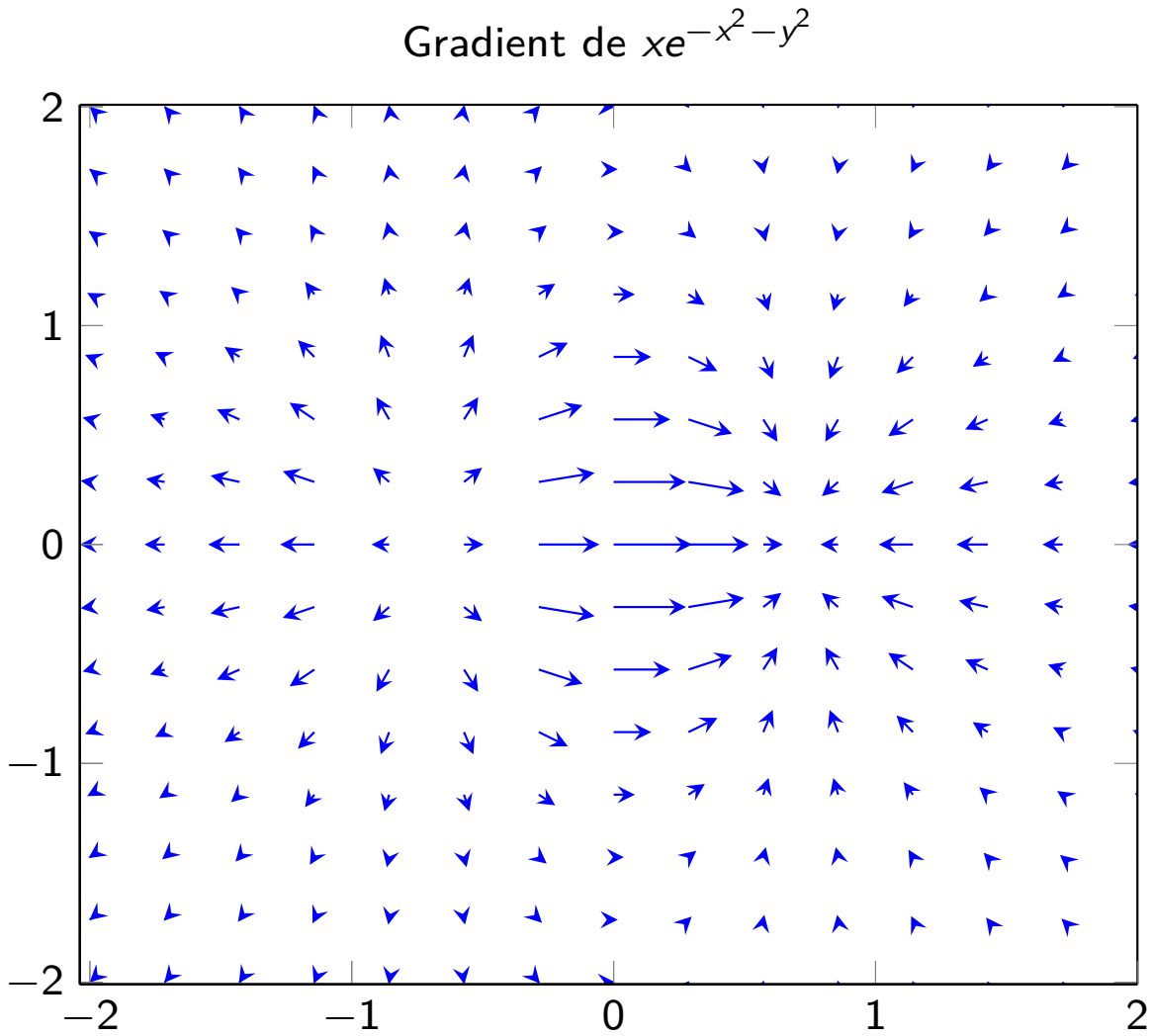
$$\vec{V} : D \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

en 3D.

Remarques

- C'est une fonction qui assigne à **chaque point** du plan ou de l'espace un **vecteur**
- L'ensemble D peut être quelconque: une région du plan, une courbe, une surface, etc.

Représentation graphique



Utilité

- Permet de représenter des **champs de force**
- On peut ensuite simuler des **phénomènes physiques**

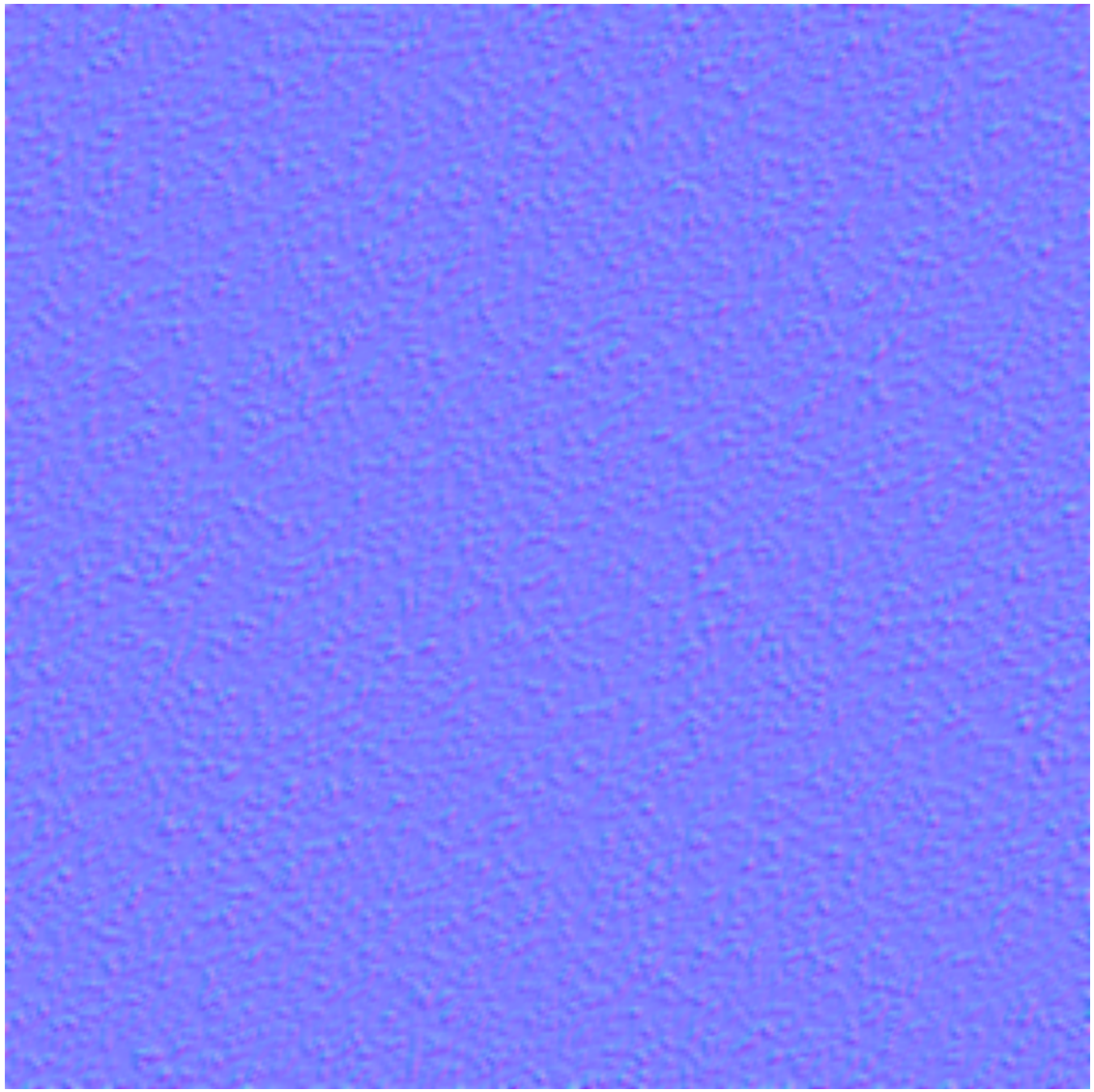
→ Vent, vortex, turbulence

→ Champ magnétique

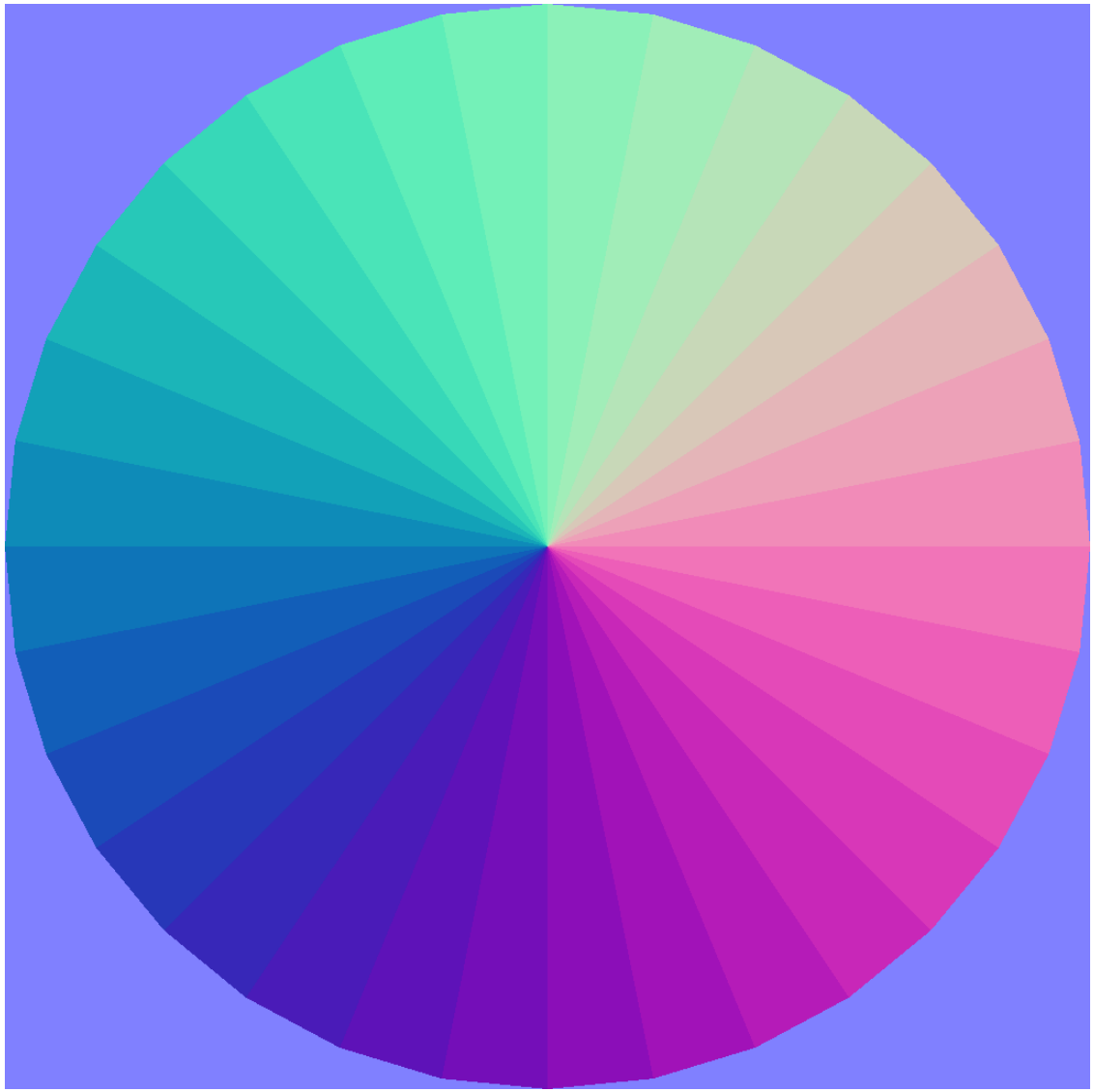
→ Voir le manuel de Blender.

- Les **normales** d'une surface sont aussi représentées par un **champ de vecteurs**

Déplacement



Cône sur plan



Mathématiquement...

- Chaque **pixel** code un **vecteur** de $\mathbb{R}^3$
- Les **composantes** du vecteur sont en correspondance avec les **couleurs**

$\rightarrow$ $x \mapsto R$

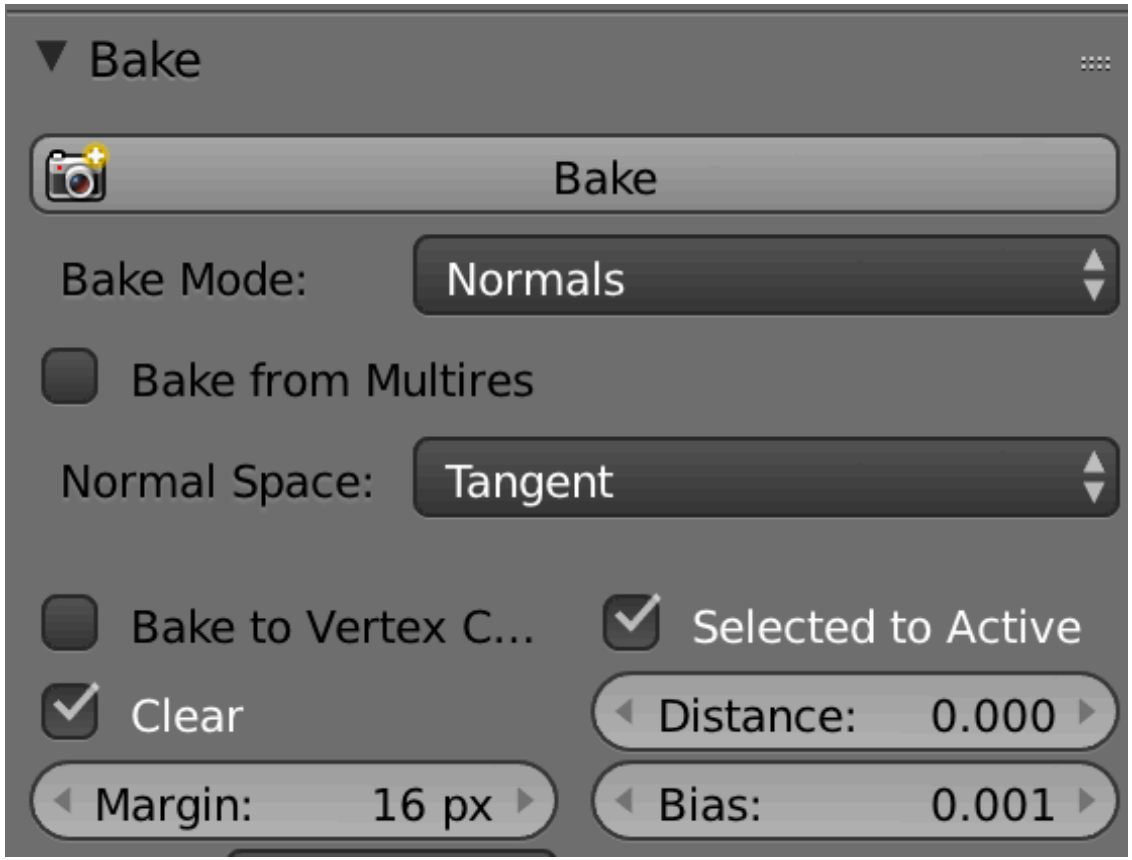
$\rightarrow$ $y \mapsto G$

$\rightarrow$ $z \mapsto B$

- Chaque composante, dans $[-1, 1]$, est représentée par un niveau de couleur, dans $[0, 1]$
- Comme la composante z ne peut jamais être **négative**, le niveau de bleu est toujours au moins 0.5

« Cuisiner » des normales

- Dans Blender, on peut « cuisiner » (*bake*) des normales
- Ne pas oublier de *développer* le modèle de basse complexité
- Et de cocher *Selected to Active*.
- Tutoriel: https://www.youtube.com/watch?v=0r-cGjVKvGw.

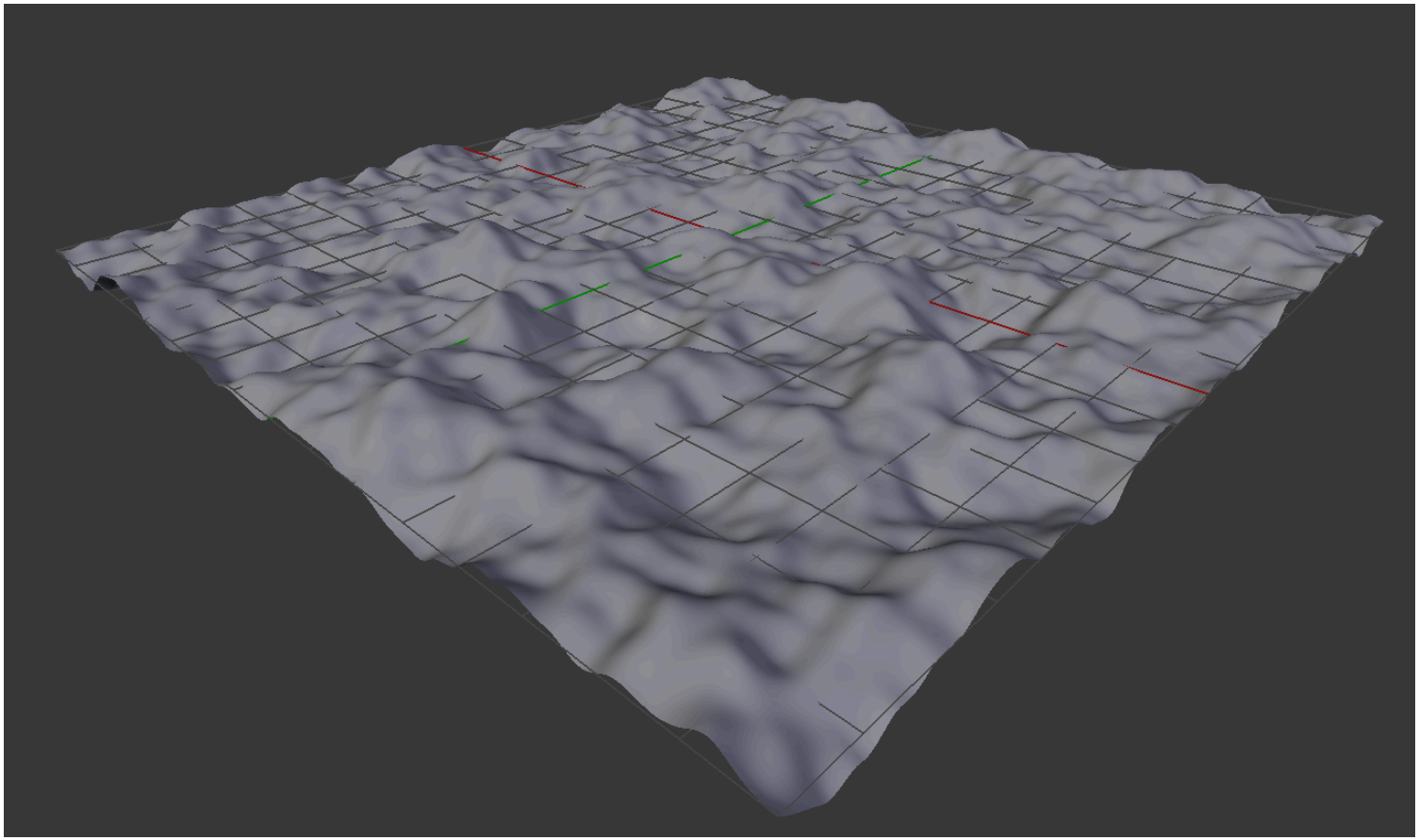


Occlusion ambiante

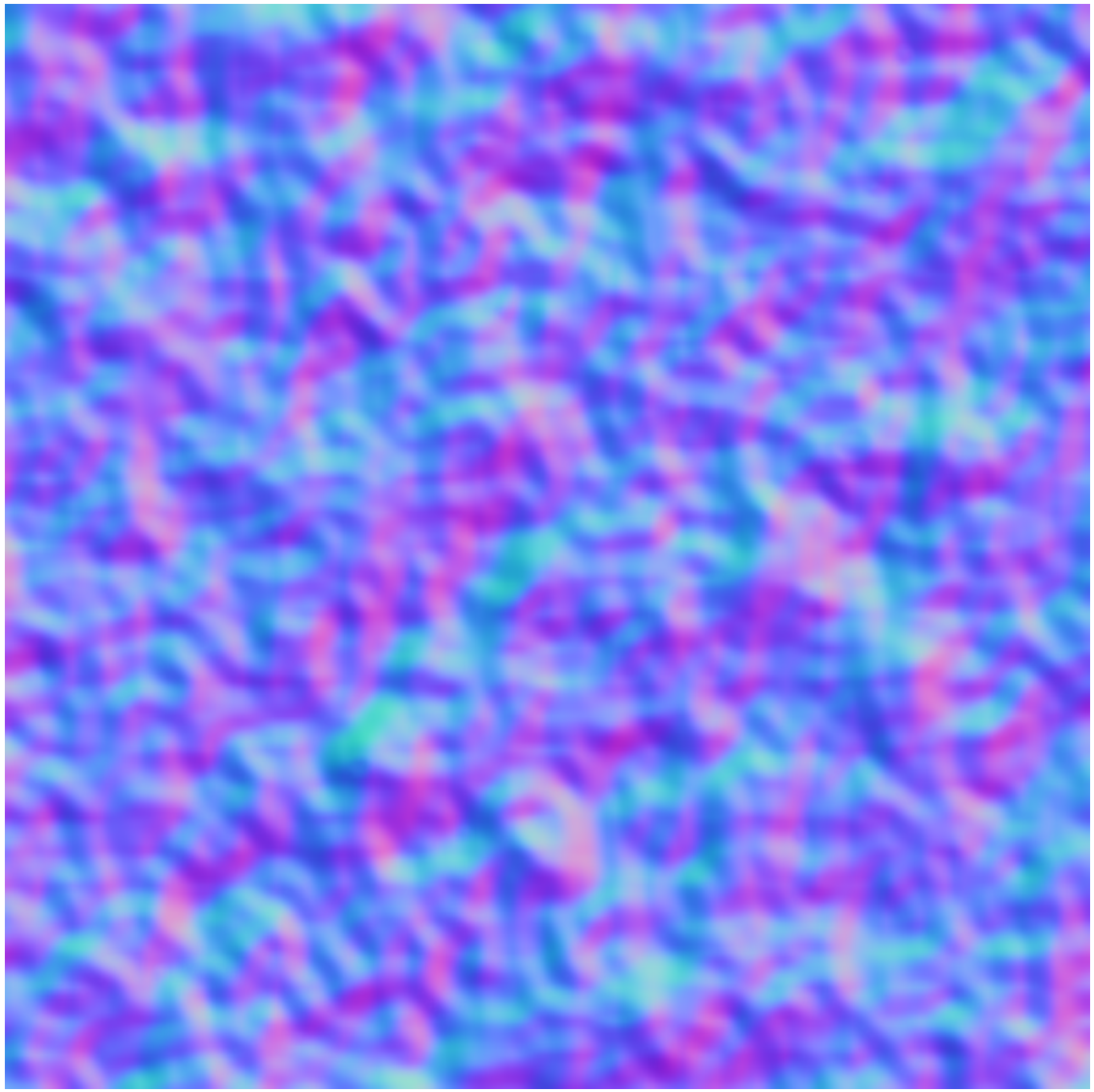
Idée générale

- L'**occlusion ambiante** est une mesure de l'**exposition** d'un point à l'**éclairage ambiant**
- C'est une valeur qui ne dépend pas des **sources de lumière**, seulement de l'**intensité** de l'éclairage ambiant
- En revanche, elle dépend des **objets** présents dans la scène
- Contrairement à l'**illumination globale**, il s'agit d'une approximation, donc elle est moins gourmande en temps de calcul
- Par conséquent, elle est souvent utilisée dans les applications en **temps réel**, en particulier les jeux vidéos

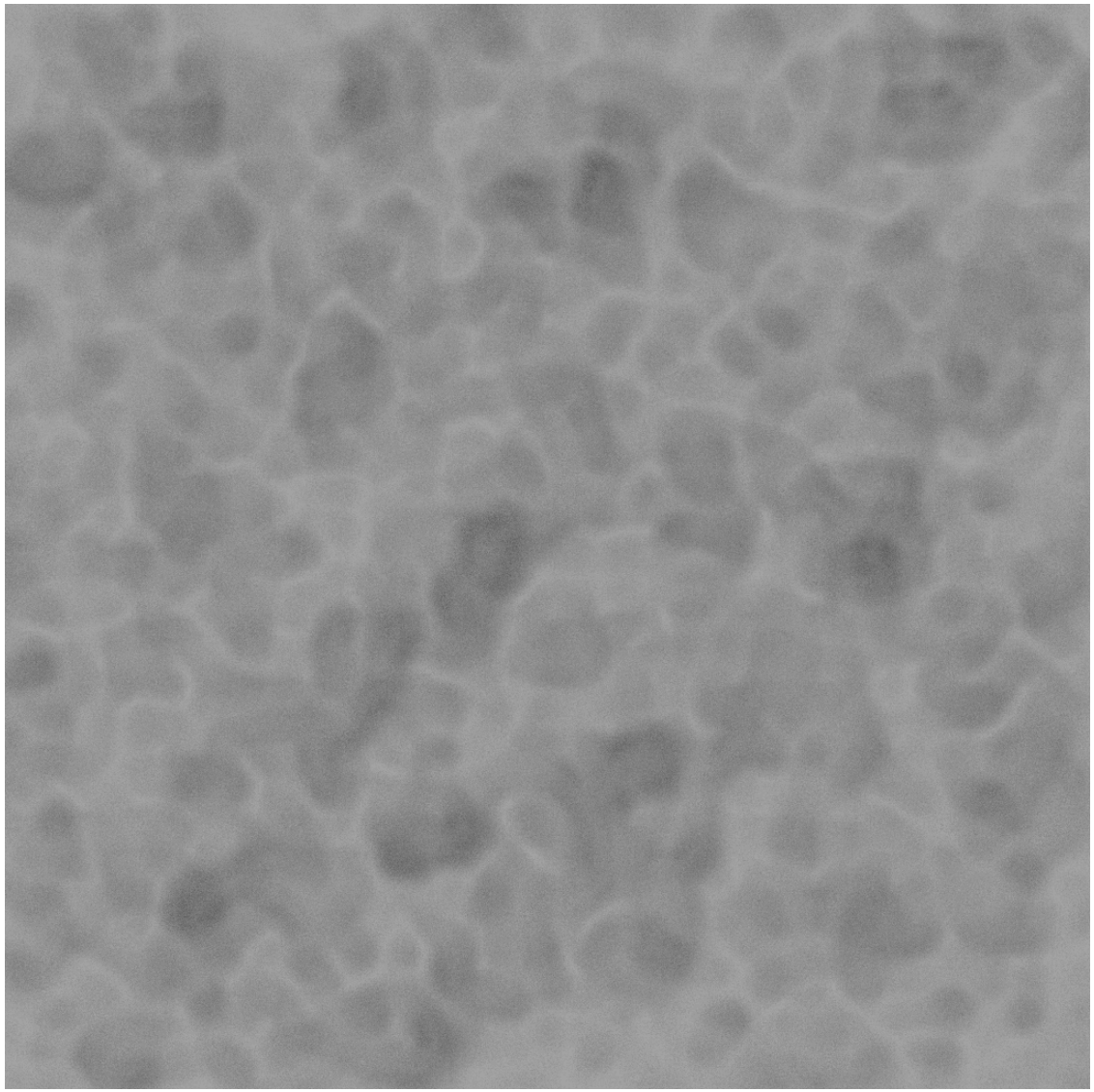
Exemple (vue en 3D)



Exemple (normales)



Exemple (occlusion ambiante)



Formule

L'occlusion ambiante d'un point p sur une surface S est donnée par

$$A_p = \frac{1}{\pi} \int_{\Omega} V_{p,\vec{\omega}} (\vec{n} \cdot \vec{\omega}) d\vec{\omega}$$

où

- $A_p \in [0, 1]$ représente l'occlusion ambiante en p;
- $\vec{n}$ représente le **vecteur normal** à S au point p;
- Ω représente l'**hémisphère unitaire** induite par $\vec{n}$;
- $\vec{\omega}$ représente un **vecteur unitaire** dans Ω.

et

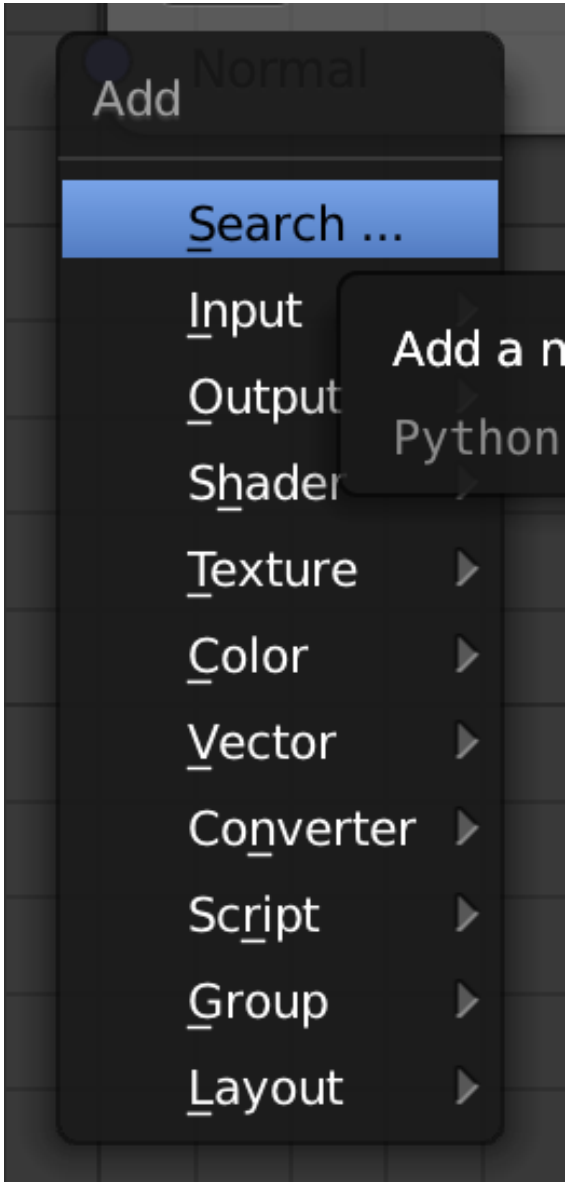
$$V_{p,\vec{\omega}} = \begin{cases} 0, & \text{si } p \text{ est bloqué dans la direction du vecteur } \vec{\omega} \\ 1, & \text{sinon} \end{cases}$$

Matériaux 3D

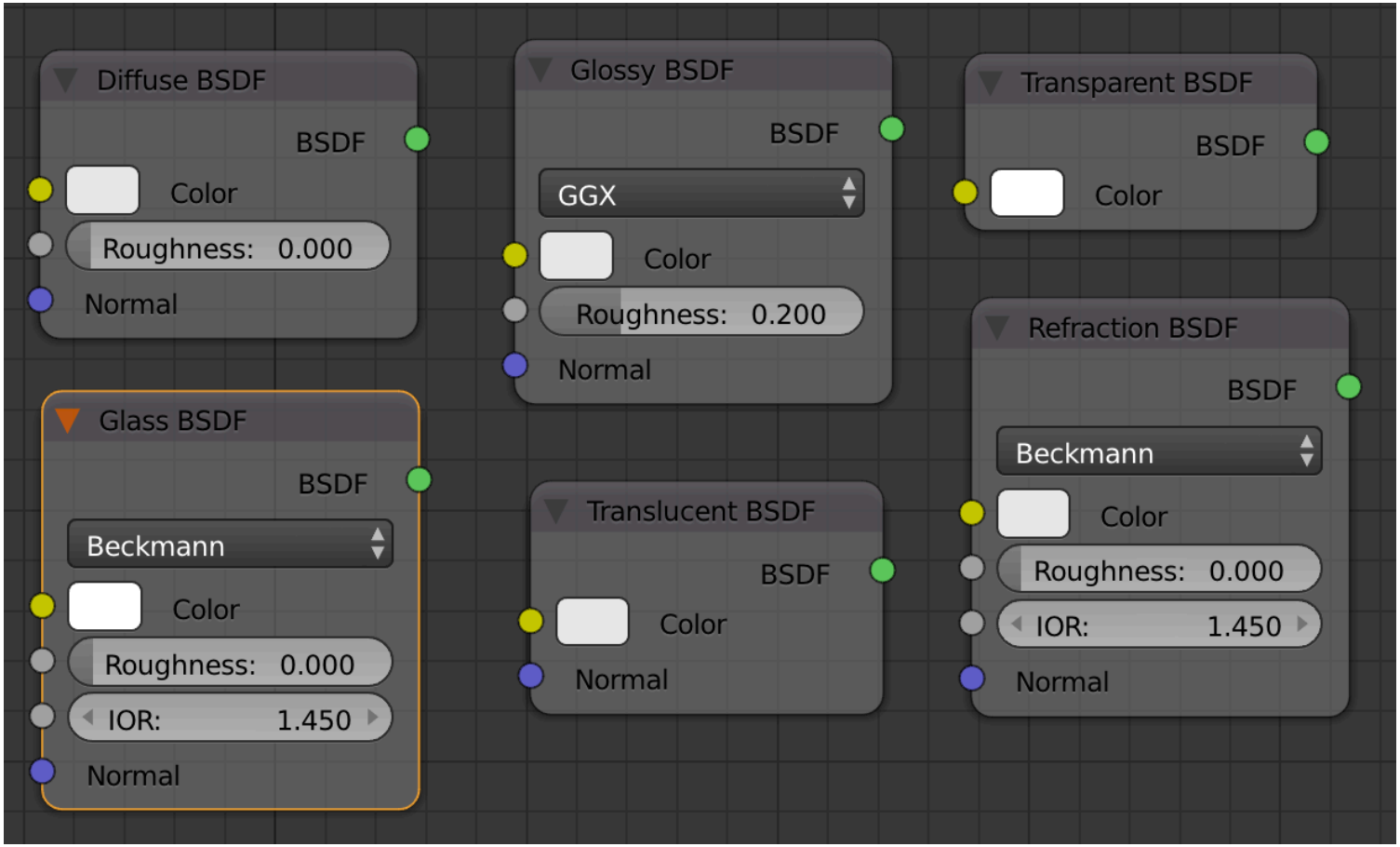
Introduction

- Jusqu'à maintenant, nous nous sommes concentrés sur des **matériaux simples**
- On fait d'abord le **développement** (*unwrap*) du modèle
- Ensuite, on lui applique une **texture**, c'est-à-dire une **image**
- Lorsqu'on souhaite avoir un grand niveau de **détails**, cependant, il est souvent trop coûteux d'avoir une géométrie complexe
- **Solution:** on utilise des applications (*maps*) simulant cette géométrie

Types de noeuds



Shaders


Diffuse BSDF
BSDF
Color
Roughness: 0.000
Normal
Glossy BSDF
BSDF
GGX
Color
Roughness: 0.200
Normal
Transparent BSDF
BSDF
Color
Glass BSDF
BSDF
Beckmann
Color
Roughness: 0.000
IOR: 1.450
Normal
Translucent BSDF
BSDF
Color
Normal
Refraction BSDF
BSDF
Beckmann
Color
Roughness: 0.000
IOR: 1.450
Normal