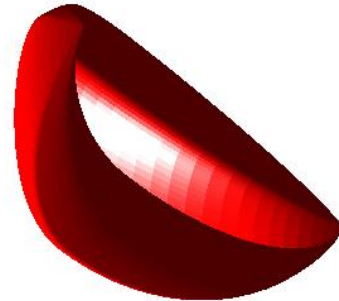


Matlab 11-12 for dummies : exercice 3

Un canoë pour le Québec...

Dans cet exercice, vous concevez des canoës : comme vous le savez certainement, la descente de rivières en canoës ou kayaks est un des atouts touristiques bien connus de notre belle région wallonne...

Pour réaliser ce problème, il n'est pas inutile de tirer profit du programme `vase.m` présenté au cours. Le programme `egg.m` de l'année passée peut aussi être une bonne source d'inspiration. L'implémentation récursive des fonctions B-splines qui y est incluse, n'est pas la plus efficace, mais elle présente l'avantage de la simplicité.



En s'inspirant du programme fourni, nous vous demandons d'écrire un programme définissant un canoë semblable à celui de la figure ci-dessus.

1. Au moyen d'un ensemble de 5×9 poids et 5×9 points de contrôle dans l'espace, définir le canoë avec des surfaces NURBS. On utilisera des surfaces de degré deux et les vecteurs de noeuds seront donnés respectivement¹ par $\mathbf{T} = [0, 0, 0, 1, 1, 2, 2, 3]$ et $\mathbf{S} = [0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5]$. Le canoë doit être défini dans une boîte $[-1, 1] \times [-1, 1] \times [-1, 2]$. L'axe du canoë est donc défini le long de l'axe Oz .
2. Inclure votre code dans une fonction `[x y z] = canoe(dt)`. Les variables `x` `y` `z` sont les trois matrices requises par l'instruction `surf`. L'argument `dt` est l'incrément utilisé pour parcourir l'espace paramétrique pour effectuer la représentation graphique. Le figure ci-dessus a été obtenue avec `canoe(0.05)`. Les trois matrices fournies seront de taille 81×41 , si on utilise un incrément `dt=0.05` pour la définition des vecteurs `t` et `s`. Attention : tous les points de la surface doivent être impérativement dans la boîte $[-1, 1] \times [-1, 1] \times [-1, 2]$.
3. Un programme `matlab3` vous est fourni pour tester votre fonction. Si votre code est correct, vous devriez un résultat approximativement semblable à celui de la figure.

```
[x y z] = canoe(0.05);
figure; ax = axes('Visible','off','PlotBoxAspectRatio',[450 350 450],...
'DataAspectRatio',[1 1 1],'CameraViewAngle',6);
view(ax,[41 -42]); hold(ax,'all');

surf(x,z,y,'Parent',ax,'BackFaceLighting','unlit','FaceColor',[1 0 0],'LineStyle','none');
light('Parent',ax,'Position',[1 0 0]);
light('Parent',ax,'Position',[10 4 3]);
```

4. Votre fonction (avec toutes les éventuelles sous-fonctions que vous auriez créées) sera incluse dans un unique fichier `canoe.m`, sans y adjoindre le programme de test fourni ! Ce fichier devra être rendu via le web avant le lundi 17 octobre à 23h59 : ce travail est individuel et sera évalué.

¹ Pour les petits futés, la question suivante est soumise à votre sagacité : pourquoi vaut-il mieux définir le dernier noeud des vecteurs \mathbf{T} et \mathbf{S} comme étant différent de l'avant-dernier noeud pour que le code fonctionne bien ? Théoriquement, cela devrait pourtant fonctionner également avec $\mathbf{T} = [0, 0, 0, 1, 1, 2, 2, 2]$ et $\mathbf{S} = [0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4]$.