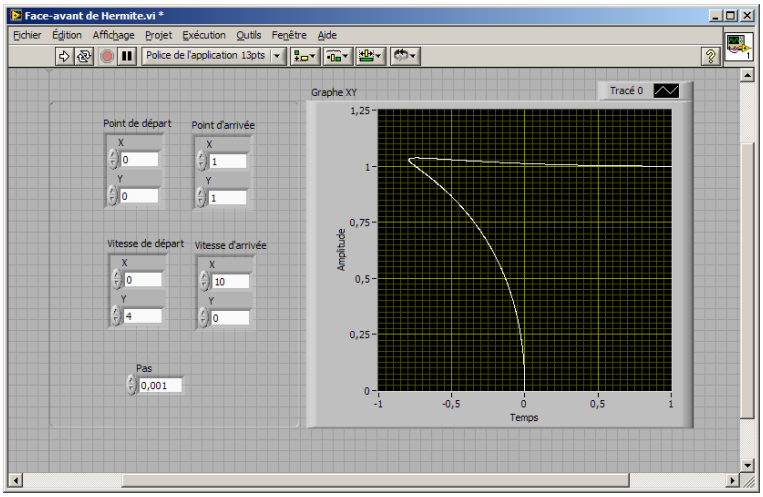
Courbes de forme libre implémentées en LabVIEW

I – Courbes de Hermite

Rédiger un programme Hermite.vi donc l'objectif sera de tracer une courbe de Hermite entre deux points P0 (Point de départ) et P1 (Point d'arrivée), en connaissant des conditions initiales V0 (Vitesse de départ) et V1 (vitesse d'arrivée) en ces points.

Voici pour info, un exemple de face avant susceptible d'être réalisée :



a – Gestion des données

Les données de points et conditions initiales (vitesse) seront gérées de façon vectorielle, sous forme de cluster à deux éléments X et Y de type double flottant.
Le pas de calcul sera un scalaire de type double flottant.

b – Calcul de la courbe de Hermite

Ecrire les fonctions suivantes :

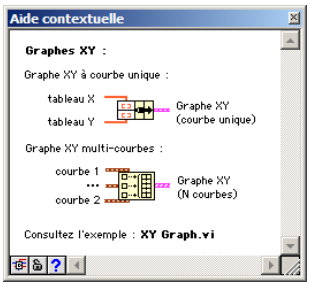
float H00 (float u) ;
float H10 (float u) ;
float H01 (float u) ;
float H11 (float u) ;

sous forme de boîte de calcul (formula node) de LabVIEW.

Le calcul des points de la courbe se fera dans une boucle FOR correctement dimensionnée, en fonction du Pas de calcul. On sortira de la boucle, deux tableaux X et Y autoindexés qui contiendront les coordonnées des points de la courbe.

c – Affichage de la courbe dans un graphe XY

Un graphe XY est documenté de la façon suivante :



Un graphe avec une seule courbe doit donc être « alimenté » par un cluster contenant les deux tableaux des coordonnées X et Y.

On peut (et c'est conseillé) rajouter dans cette fonction l'affichage dans d'autres couleurs des points de données et des conditions aux limites. Bien réfléchir ...

II – Courbes B-Splines

Rédiger un programme BSPLINE.VI donc l'objectif sera de tracer une courbe B-Spline définie par un ensemble de points de contrôle P(i) i = 0, …, NC-1. dans un graphe XY. On pourra se baser sur le VI précédent (en le sauvegardant auparavant)

Les données seront gérées de façon analogue, à l'exception que les points de contrôle seront stockés dans un tableau de clusters.

a – Lecture des données

Si dans un premier temps, les données pourront être interactivement entrées dans un tableau, on pourra ensuite imaginer de les lire dans un fichier au format texte. Voir en ce sens les outils de lecture de fichiers « tableurs » (spreadsheets) sous LabVIEW

b – Calcul de la courbe B-Spline

Ecrire la fonction suivante, sous forme de boite de calcul :

```
float B (float u, int k) ;
```

qui est l'implémentation des 4 fonctions Spline de base. Si *k* vaut 0, B retourne B0(*u*), si *k* vaut 1, B retourne B1(*u*), …..

Ecrire la structure de génération de la B-spline qui calcule pour un paramètre *i* compris entre 1 et NC-2 et pour un paramètre *u* compris entre 0 et 1 les points *x* et *y* correspondants sur la courbe à calculer.

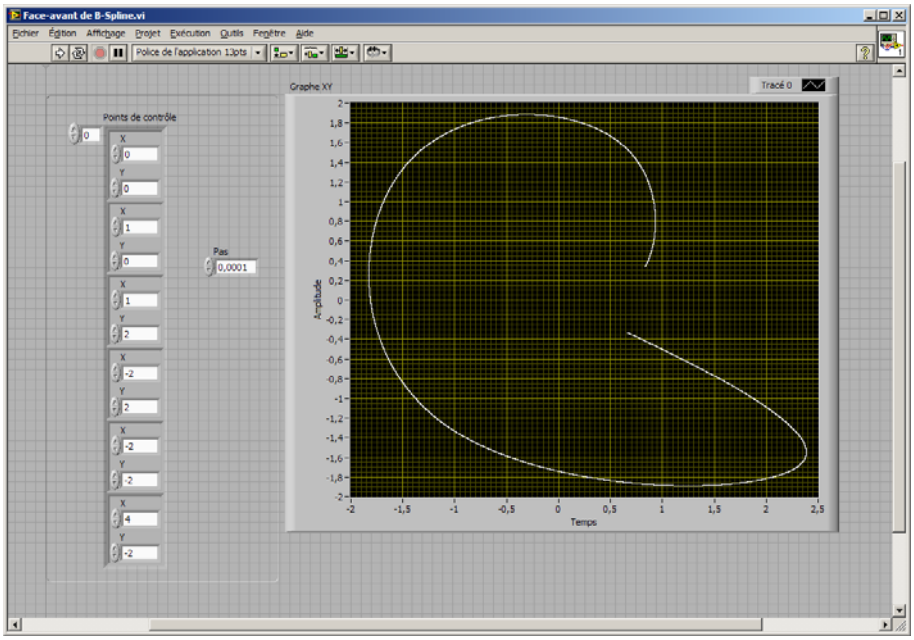
On rappelle que le calcul des *x* et *y* dans un segment *i* donné de la B-spline peut être illustré (selon le formalisme C) par:

```
for (u=0; u<1; u+=Pas)
{
        x = 0 ; y = 0 ;
        for (k=0; k<=3; k++)
        {
                x(u) += CX[i+k-1]*B(u,k) ;
                y(u) += CY[i+k-1]*B(u,k) ;
        }
}
```

Indications : On pourra gérer les « segments courbes» i de la b-spline selon une structure FOR, à l'intérieur de laquelle on positionnera une boucle WHILE de discrétisation de ce segment selon un Pas. Le reste de la programmation est laissée à votre appréciation.

c – Affichage de la courbe dans un graphe XY

Une fois la courbe B-spline calculée, le graphe XY est alimenté de façon similaire à l'exercice sur les courbe de Hermite.



On peut (et c'est conseillé) rajouter dans cette fonction l'affichage dans d'autres couleurs des points de contrôle.

BON COURAGE