Université Pierre et Marie Curie
LM 383
Année universitaire 2007-2008

Equations Differentielles, méthodes de résolution

T. Capricelli et **D. Yakoubi**

Première séance SCILAB
Représentation et approximation de fonctions d'une variable: corrigé

1. **Interpolation par des splines**
   On rappelle que l'on utilise les vecteurs

```
x=[0:0.02:1]'
xp=[0:0.2:1]'
```

   Le calcul (des dérivées aux points `xp`) de la spline cubique est donné par

```
d=splin(xp,g(xp))
```

   En interpolant, les valeurs de la spline correspondant aux abscisses `x` sont alors données par

```
f=interp(x,xp,g(xp),d)
```

   Représentation graphique

```
xbasc()
plot2d(x,g(x),style=5)
plot2d(x,f,style=4)
```

2. **Interpolation de Lagrange**
   En lisant la représentation formelle des coefficients $f[x_0,..,x_k]$ colonne par colonne, on aboutit au pseudo-algorithme :

```
initialisation :

pour i allant de 0 a n
 c(i):=h(x(i))
finpour

etapes suivantes :

pour j allant de 1 a n
    pour i allant de n a j
     c(i):=(c(i)-c(i-1))/(x(i)-x(i-j))
    finpour
finpour
```

   Il suffit alors de décaler les indices pour obtenir la fonction Scilab suivante

```
function [c]=DiffDiv(x,h)
 n=length(x)-1
 c=0*x
 for i=1:n+1
  c(i)=h(x(i))
```

```
 end
 for j=2:n+1
  for i=n+1:-1:j
   c(i)=(c(i)-c(i-1))/(x(i)-x(i-j+1))
  end
 end
endfunction
```

Algorithme de Horner adapté :

```
function [y]=HHorner(c,x,t)
 y=c(n+1)
 for i=n:-1:1
  y=c(i)+y*(t-x(i))
 end
endfunction
```