

ANALYSE NUMÉRIQUE - TPs SCILAB

Proposition d'exercices pour la session 2 de juin 2011

Exercice 1 (3 points)

On considère le texte suivant :

We can use Scilab procedures to deal with polynomials. The polynomial $a_1 + a_2x + \dots + a_mx^{m-1}$ can be represented in SCILAB symbolically.

In the following example, $-4 - 3x + x^2$ is represented by the vector of coefficients $[-4 \ -3 \ 1]$ and by a polynomial p . We setup the polynomial with the command

```
v = [-4,-3,1]
p=poly(v,'x','coeff')
```

We can also build up a polynomial in a more natural way.

```
x=poly(0,'x') // Seed a Polynomial using variable 'x'
p=x^2-3x-4     // Represents x^2-3x-4
```

There are many useful functions that can be applied to polynomials. We can find the roots (numerically) of a polynomial, evaluate a polynomial or find its derivative.

```
z = roots( p )
z(1)^2-3z(1)-4 // Two ways to evaluate the poly.
horner(p,z(1)) // at the first root.
derivat(p)     // Calculate the derivative of the polynomial
```

1. Déterminer l'erreur récurrente qui se cache dans les lignes de code de ce texte.
2. Présenter sous la forme d'un pseudo-code la méthode de Bernoulli.
3. Cette méthode permet-elle de déterminer toutes les racines du polynôme p ?

Exercice 2 (2 points)

Déterminer la méthode codée par la fonction `main` suivante (implémentée sous SCILAB), en ayant au préalable détaillé ses étapes.

```
function y=main(b,h)
// h représente le pas, c'est-à-dire la longueur d'une subdivision
n=floor(b/h) ;
// la commande floor calcule la partie entière de b/h et donne donc le nombre
de subdivisions de l'intervalle
x=0 :h :(n-1)*h ;
y(1)=1 ;
for i=2 :n ;y(i)=y(i-1)+y(i-1)*h ;end
plot2d(x,y),plot2d(x,exp(x),style=5)
endfunction
main(3,0,1)
```