CALCUL DE RACINES EN « *J* »

R. Coquidé

Racines d'un polynôme

(expression usuellement utilisée pour « solutions d'une équation polynomiale »)
Le langage J est pourvu d'un outil puissant pour calculer les racines d'un polynôme : le verbe « p. ».
Il est possible de définir un polynôme par ses racines.
Par exemple le polynôme suivant défini par ses facteurs
3(x-1)(x-2)(x+1.5)(x-0.5)(x-1.2)
sera créé par

```
[pr =. 3;1 2 _1.5 0.5 1.2 NB. On peut remarquer le
                          NB. facteur 3 et la «mise en boîte »
                          NB. de la variable pr (pro-nom)
```

3	1 2 _1.5 0.5 1.2

Avec le verbe « p. », calcul des coefficients du polynôme :

```
[pc =. p. prNB. A partir des racines
5.4 _19.8 19.05 1.95 _9.6 3   NB. Ici, pas de« boîte »
```

Avec le même verbe « p. » calcul des racines à partir des coefficients :

```
p. pc NB. rangées dans l'ordre des modules décroissants
```

3	2 _1.5 1.2 1 0.5

Les racines complexes sont acceptées … et traitées…

```
[pr =. 2 ; 0j0.2 _1 0j_0.2 1j2 1j_2 2
```

2	0j0.2 _1 0j_0.2 1j2 1j_2 2

```
[pc =. p. pr
_0.8 _0.08 _19.6 _2.24 10.08 _6 2
   p. pc
```

2	1j2 1j_2 2 _1 0j0.2 0j_0.2

… les coefficients complexes aussi

```
   [pr =. 2j_1 ; 1j0.1 _1 0j_0.2 1j2 1j_2 _2j_0.5
```

2j_1	1j0.1 _1 0j_0.2 1j2 1j_2 _2j_0.5

```
[pc =. p. pr
_0.55j_4.6 _23.13j2.39 _0.57j5.01 22.25j_6.35 _1.16j_2.17
    0.6j1.2 2j_1

p. pc
```

2j_1	1j2 1j_2 _2j_0.5 1j0.1 _1j0 0j_0.2

Solution de f(x) = 0   où f(x) n'est pas polynomiale
(on suppose connu un intervalle [a , b] contenant une solution et une seule : x).
Il est permis de s'inspirer de l'algorithme de NEWTON (ou de la tangente):
x_{n+1} = x_n − f(x_n)/f'(x_n)   qui calcule x_{n+1}, approximation meilleure que x_n
NEWTON =.]-([. %].)   NB. conjonction
Usage : x_{n+1} =.   (f NEWTON fp) x_n
x_{n+k} =. ((f NEWTON fp)^ : k) x_n
Ici, on utilise la « puissance fonctionnelle » k (entier positif ou nul).
La suite récurrente (x_n) converge (quand tout se passe bien !) vers x, racine de l'équation f(x) = 0 ; fp(x) est la dérivée de f(x)

Exemple : pour résoudre x = cos(x) ou f(x) = 0 avec f(x) = x − cos(x),
écrivons :
f =. −2&o.   NB. verbe représentant la fonction x-cos(x)
fp =. 1: + 1&o.   NB. verbe représentant sa dérivée première
Partons de x_0 = 1.5NB. Si nous utilisons

NB. x_{n+1} = cos(x_n)
,. (2&o.)^:(i.20) 1.5
1.5
0.070737201667703
0.997499167206586
0.542404992339220
0.856469708947328
0.655108801780784
0.792981645797353
0.701724168276573
0.763730311290859
0.722261082134520
0.750312885709986
0.731475558034674
0.744189586630583
0.735637098348715
0.741403372909269
0.737521554461599
0.740137474793432
0.738375854160239
0.739562726170268
0.738763336576956
etc…

Convergence lente
(d'ordre 1)

NB. Si nous utilisons
NB. la conjonction NEWTON
,. (f NEWTON fp)^:(i.6)   1.5
1.5
0.784472397719411
0.739518709832052
0.739085174705196
0.739085133215161
0.739085133215161
Convergence beaucoup
plus rapide
(d'ordre 2).

On a défini et exécuté

```
P =. 9!:11
P   15
```

pour obtenir 15 chiffres affichés.

```
,. (f  NEWTON  fp)^:(_)    1.5
0.739085133215161
```

Utilisation de la « puissance fonctionnelle » infinie.

A suivre!