Evaluation expérimentale de la rapidité de convergence d'une suite

Groupe algorithmique de l'IREM d'Aix-Marseille
Henri ROLAND 2010-2011

Présentation générale

Considérons une suite (u_n) qui converge vers l

On dit que la convergence est d'ordre p lorsque $u_{n+1} - l$ est equivalent à $A\,(u_n - l)^p$ où A est dans l'intervalle]-1,1[.

Si on pose $v_n = u_n - l$, on a alors $\ln\left(\frac{v_{n+2}}{v_{n+1}}\right)$ qui est équivalent à $p \ln\left(\frac{v_{n+2}}{v_n}\right)$ et $p = \lim_{n\to\infty} \frac{\ln\left(\frac{v_{n+2}}{v_{n+1}}\right)}{\ln\left(\frac{v_{n+1}}{v_n}\right)}$

p étant connu on peut calculer $A = \lim_{n\to\infty} \frac{v_{n+1}}{v_n^{\,p}}$

```
> ordre:=proc(u,nmax)
    seq(abs(ln((u[i+2]-u[nmax])/(u[i+1]-u[nmax]))/
    abs(ln((u[i+1]-u[nmax])/(u[i]-u[nmax]))),i=1..nmax-3)
  end proc:
```

```
> coefficient:=proc(u,p,nmax)
    seq(abs((u[i+1]-u[nmax])/(u[i]-u[nmax])^p),i=1..nmax-2);
  end proc:
```

La méthode de la corde

La procédure suivante calcule la liste des termes de la suite définie par $u_0 = b$ et $u_{n+1} = a - \frac{(u_n - a)\,\mathrm{f}(a)}{\mathrm{f}(u_n) - \mathrm{f}(a)}$ pour a et b donnés

Voir le document intitulé La méthode de la corde
auteur : Henri ROLAND Groupe algorithmique de l'IREM d'Aix-Marseille

```
> corde:=proc(f,a,b,n)
  local bb,ya,i,L;
  bb:=b;L:=bb;
    ya:=f(a);
    for i from 1 to n do
      bb:=a-(bb-a)*f(a)/(f(bb)-f(a));
      L:=L,bb;
    end do;
    [L]
  end proc:
```

```
> f:=x->x^3-2:
> Digits:=300:interface(displayprecision=10):u:=corde(f,2.,1.,42);
```

u := [1.0000000000, 1.1428571429, 1.2096774194, 1.2388370021, 1.2511598712, 1.2562955295, 1.2584233391, 1.2593027847, 1.2596659013, 1.2598157668, 1.2598776087, 1.2599031258, 1.2599136544, 1.2599179985, 1.2599197909, 1.2599205304, 1.2599208356, 1.2599209615, 1.2599210134, 1.2599210348, 1.2599210437, 1.2599210473, 1.2599210488, 1.2599210495, 1.2599210497, 1.2599210498, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499]

Rapidité de convergence de la méthode de la corde

```
> c1:=ordre(u,43);
```

$c1$:= 1.0603996818, 1.0266373560, 1.0113076790, 1.0047214916, 1.0019577536, 1.0008094239, 1.0003342502, 1.0001379595, 1.0000569301, 1.0000234907, 1.0000096925, 1.0000039992, 1.0000016501, 1.0000006808, 1.0000002809, 1.0000001160, 1.0000000481, 1.0000000203, 1.0000000095, 1.0000000066, 1.0000000094, 1.0000000199, 1.0000000470, 1.0000001135, 1.0000002750, 1.0000006664, 1.0000016150, 1.0000039143, 1.0000094871, 1.0000229941, 1.0000557334, 1.0001350987, 1.0003275505, 1.0007945597, 1.0019297994, 1.0047011892, 1.0115377983, 1.0288480777, 1.0757704101, 1.2318601562

```
> c2:=coefficient(u,1.,43);
```

$c2$:= 0.4503825569, 0.4291983055, 0.4196362323, 0.4155358975, 0.4138165077, 0.4131023023, 0.4128068005, 0.4126847366, 0.4126343492, 0.4126135554, 0.4126049751, 0.4126014348, 0.4125999741, 0.4125993714, 0.4125991227, 0.4125990201, 0.4125989777, 0.4125989602, 0.4125989527, 0.4125989493, 0.4125989468, 0.4125989434, 0.4125989362, 0.4125989190, 0.4125988775, 0.4125987771, 0.4125985337, 0.4125979437, 0.4125965140, 0.4125930487, 0.4125846498, 0.4125642927, 0.4125149483, 0.4123953197, 0.4121051782, 0.4114007834, 0.4096865456, 0.4054900927, 0.3950674337, 0.3682229433, 0.2920849889

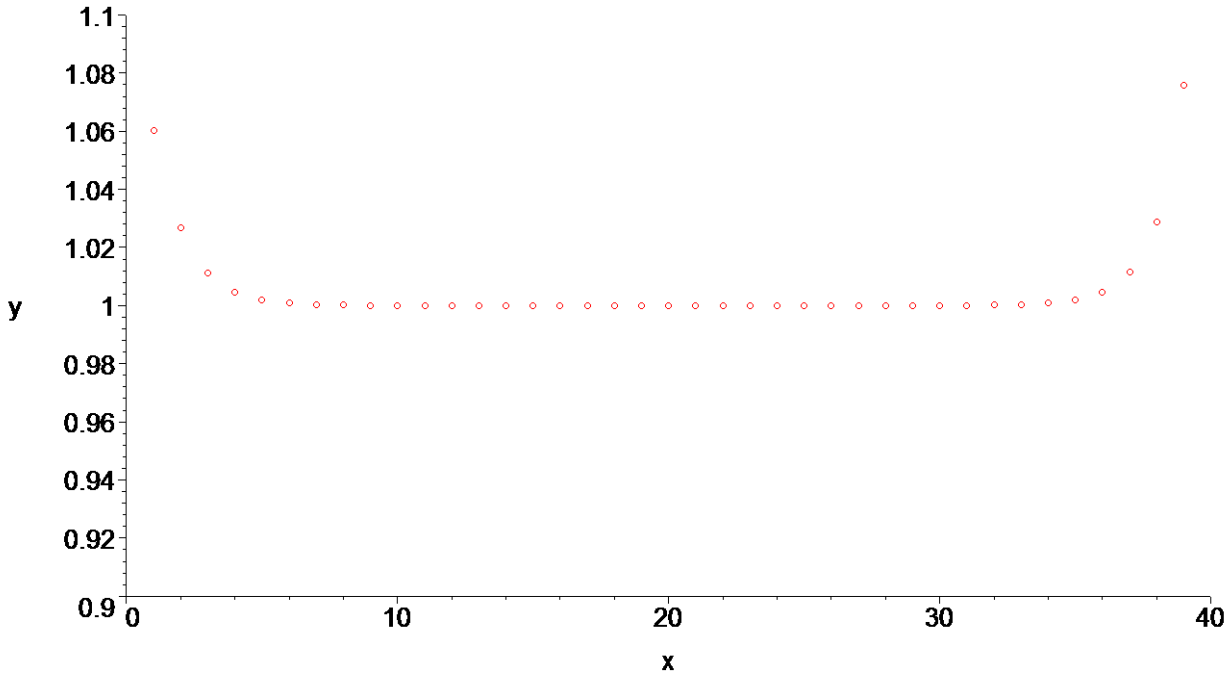
Remarques

Le coefficient A dépend de la fonction f choisie. Son calcul permet de vérifier par sa stabilité relative que la conjecture faite sur l'ordre est correcte.

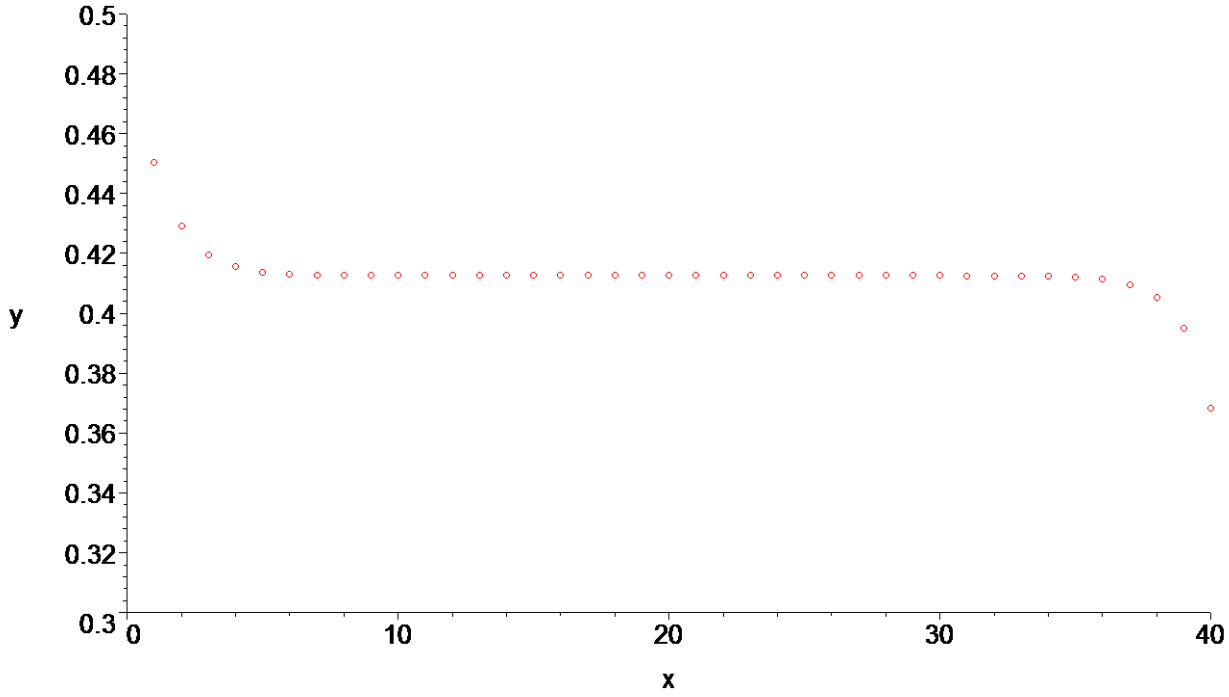
Malgré les 300 chiffres significatifs utilisés pour le calcul le quotient de deux nombres infiniments petits produit des erreurs qui font que les derniers termes de chaque liste doivent être considérés comme erratiques.

Les deux graphiques suivants donnent un aperçu de la relative stabilité de la valeur approchée de l'ordre et du coefficient A.

```
> plot([seq([n,c1[n]],n=1..40)],x=0..40,y=0.9..1.1,style=point,sym
  bol=circle,symbolsize=12);
```




```
> plot([seq([n,c2[n]],n=1..40)],x=0..40,y=0.3..0.5,style=point,sym
  bol=circle,symbolsize=12);
```



Conclusion

La convergence de la méthode de la corde est linéaire :$u_{n+1} - l$ est equivalent à $A\,(u_n - l)$ où A est voisin de 0.4126 sur cet exemple.

La méthode de Newton

La procédure suivante calcule la liste des termes de la suite définie par $u_0 = a$ et $u_{n+1} = u_n - \dfrac{f(u_n)}{g(u_n)}$

pour a donné où g désigne la fonction dérivée de f.
Voir le document intitulé La méthode de Newton
Auteur : Henri ROLAND Groupe algorithmique de l'IREM d'Aix-Marseille

```
> newton:=proc(f,a,n)
  local aa,g,i,L;
    aa:=a;
    L:=aa;
    g:=D(f);
    for i from 1 to n do
      aa:=aa-f(aa)/g(aa);
      L:=L,aa;
    end do;
    [L]
  end proc:
```

```
> v:=newton(f,2.,42);
```

$v := [2.0000000000, 1.5000000000, 1.2962962963, 1.2609322247, 1.2599218606, 1.2599210499]$

Rapidité de convergence de la méthode de Newton

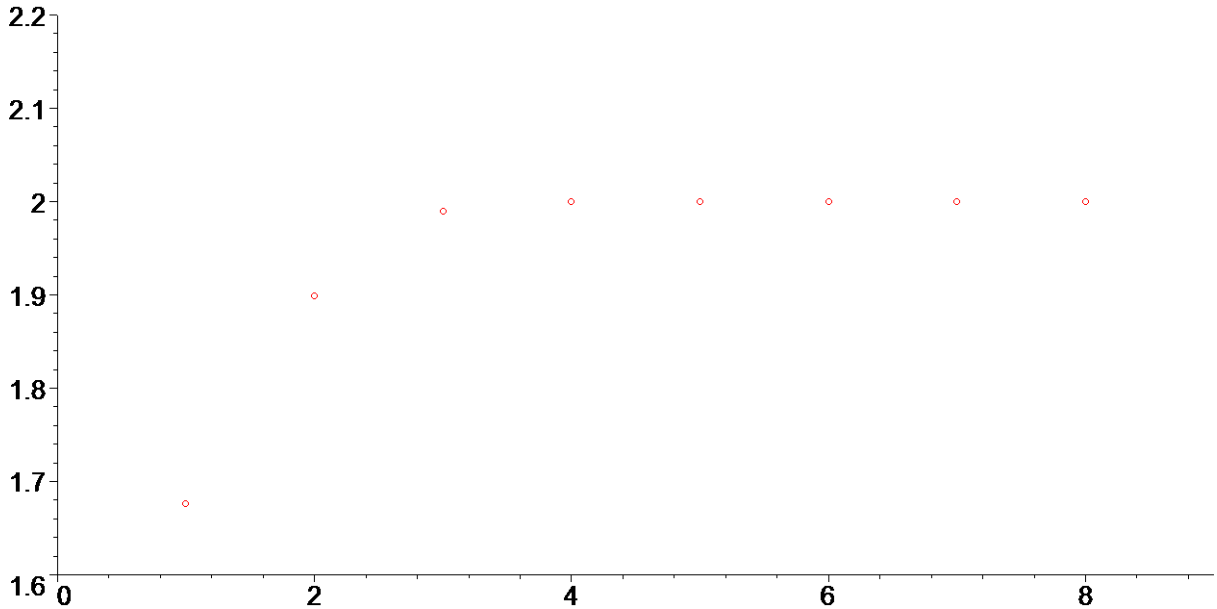
```
> c3:=ordre(v,12);
```

$c3 := 1.6762281907, 1.8985824154, 1.9897313700, 1.9998500813, 1.9999999398, 2.0000000000, 2.0000000000, 2.0000000000, \text{Float}(\infty)$

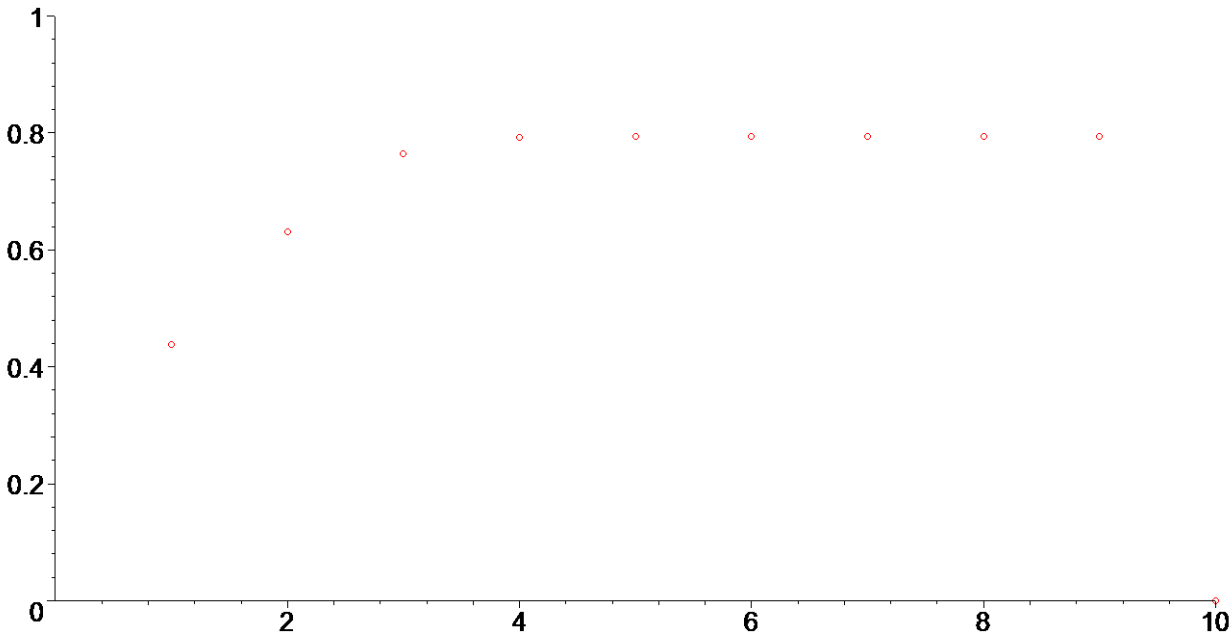
```
> c4:=coefficient(v,2.,12);
```

$c4 := 0.4383267542, 0.6310994148, 0.7642129103, 0.7928520436, 0.7936998451, 0.7937005260, 0.7937005260, 0.7937005260, 0.7937005260, 0.0000000000$

```
> plot([seq([n,c3[n]],n=1..9)],view=[0..9,1.6..2.2],style=point,symbol=circle,symbolsize=12);
```




```
> plot([seq([n,c4[n]],n=1..10)],view=[0..10,0..1],style=point,symbol=circle,symbolsize=12);
```



Conclusion

La convergence de la méthode de la Newton est quadratique : $u_{n+1} - l$ est equivalent à $A\,(u_n - l)^2$ où A est voisin de 0.7937 sur cet exemple.

La méthode avec dérivée approchée

La procédure suivante calcule la liste des termes de la suite définie par $u_0 = a$ et $u_1 = b$ et $u_{n+1} = u_n - \frac{(u_n - u_{n-1})\,\mathrm{f}(u_n)}{\mathrm{f}(u_n) - \mathrm{f}(u_{n-1})}$ pour a et b donnés .

```
> newton2:=proc(f,a,b,n)
  local aa,bb,cc,ya,yb,i,L;
    aa:=a;ya:=f(aa);
    bb:=b;yb:=f(bb);
    L:=aa,bb;
    for i from 2 to n do
      cc:=bb-(bb-aa)*f(bb)/(f(bb)-f(aa));
      L:=L,cc;
      aa:=bb;ya:=yb;
      bb:=cc;yb:=f(cc);
    end do;
    [L]
  end proc:
```

Rapidité de convergence de la méthode approchée de Newton

```
> w:=newton2(f,2.,1.,16);
```

$w := [2.0000000000, 1.0000000000, 1.1428571429, 1.2899408284, 1.2570011055, 1.2598524654, 1.2599212091, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, 1.2599210499, \mathrm{Float}(\mathrm{undefined})]$

```
> c5:=ordre(w,16);
```

$c5 := 0.2408917955, 4.2921600581, 1.1424868180, 0.9590310239, 1.8209764665, 1.5091348562, 1.5408316305, 1.6301716339, 1.6106212383, 1.6134506367, 1.6187021270, 1.6176200082, \mathrm{Float}(\infty)$

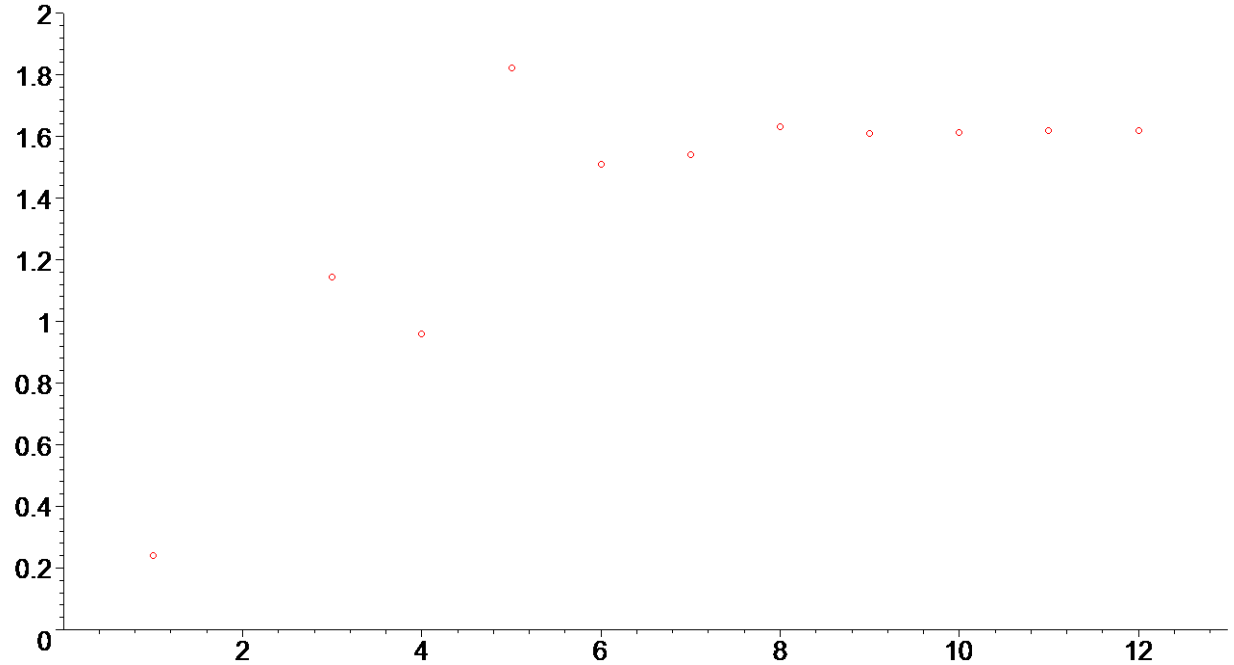
```
> Phi:=evalf((sqrt(5)+1)/2);
```

$$\Phi := 1.6180339887$$

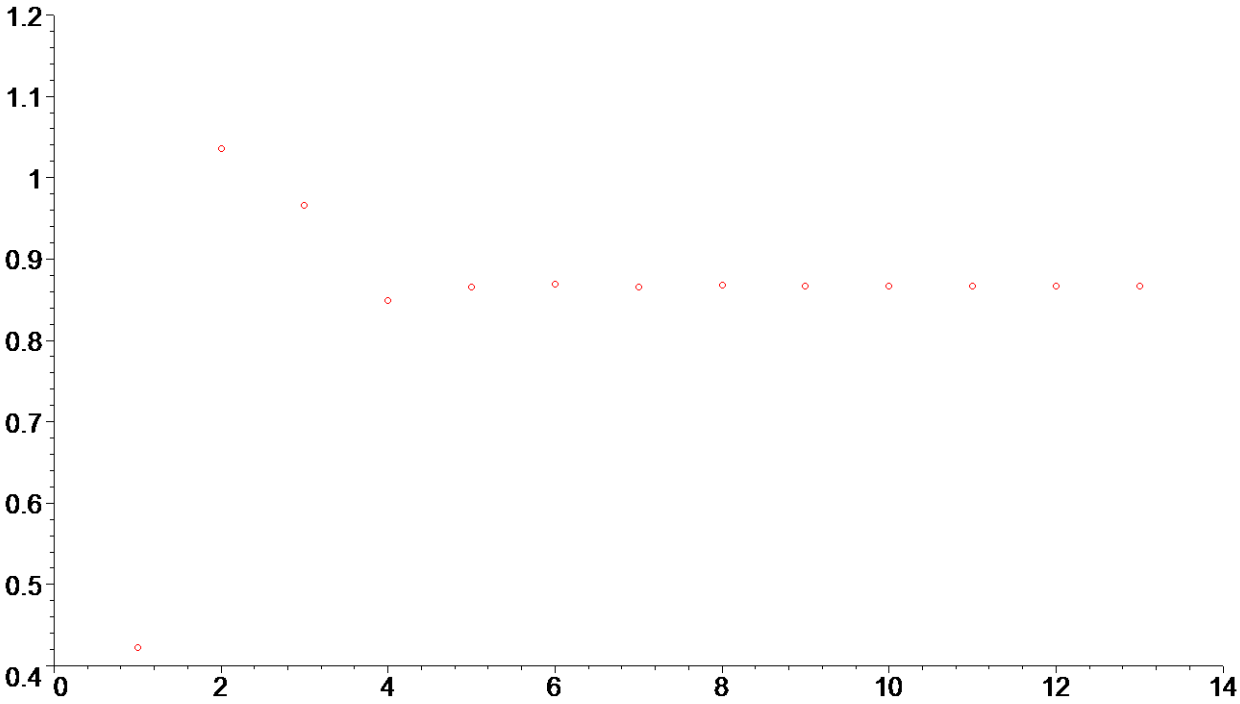
```
> c6:=coefficient(w,Phi,16);
```

$c6 := 0.4230132859, 1.0356891660, 0.9654520498, 0.8491433673, 0.8656370256, 0.8691051269, 0.8656209813, 0.8677411580, 0.8664302776, 0.8672402123, 0.8667395558, 0.8670489444, 0.8668577187, 0.0000000000$

```
> plot([seq([n,c5[n]],n=1..13)],view=[0..13,0..2],style=point,symb
  ol=circle,symbolsize=12);
```




```
> plot([seq([n,c6[n]],n=1..14)],view=[0..14,0.4..1.2],style=point,
  symbol=circle,symbolsize=12);
```




```
>
```

Conclusion

On constate donc que la méthode de Newton approchée (ou méthode de la sécante) a une vitesse de convergence d'ordre Φ où Φ désigne le nombre d'or.