J.-P. Grivet

Retour au site web

Exercice 2-8 : Développement limité et représentation en fraction continue

a) La fonction $y = \operatorname{argth} x$ a pour dérivée $y' = 1/(1-x^2)$, dont le développement limité, au voisinage de l'origine, s'écrit

$$y' \sim 1 + x^2 + x^4 + x^6 + \cdots$$

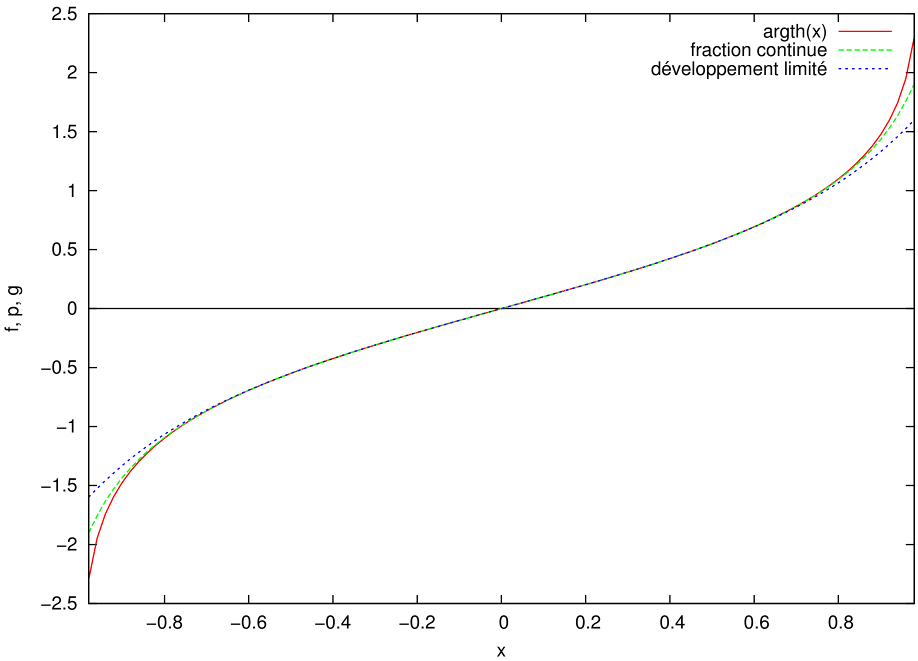
En intégrant terme à terme, on trouve le développement de $\operatorname{argth} x$, tronqué à l'ordre 7 :

$$p = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7}.$$

b) D'autre part, en chassant les dénominateurs de la fraction continue, on obtient

$$g = \frac{5x}{3}\frac{21 - 11x^2}{35 - 30x^2 + 3x^4}.$$

Ces trois fonctions sont représentées sur la figure ci-contre.



c) Les coefficients de la fraction continue s'écrivent, selon la notation du texte :

$$b_0 = 0; \quad a_1 = x; \quad b_1 = 1; \quad a_2 = -x^2; \quad b_2 = 3$$

et pour $n \geq 2$

$$a_n = -(n-1)^2 x^2 \quad ; \quad b_n = 2n - 1.$$

On déduit de la relation de récurrence les cas particuliers $A_1 = x$ et $B_1 = 1$. Il est assez facile de programmer ces relations de récurrence. Cependant, vous vous apercevrez que la convergence devient très lente, voire impossible, au voisinage de l'asymptote verticale ($x = 1$). Une petite modification de l'algorithme rend celui-ci beaucoup plus stable. Comme les relations de récurrence sont homogènes, on peut, à chaque itération, diviser l'ensemble des variables qui interviennent par b_n. C'est ce qui est fait dans le programme ci-dessous. Pour calculer d'autres fonctions, il faudrait s'assurer que $b_n \neq 0$.

```
//Exercice 2-8: calcul de argth(x) d'après la fraction continue
// et l'algorithme de Wallis
   TOL = 1.0e-12; NMAX = 100;
   anm2 = 1.0; anm1 = 0.0;
   bnm2 = 0.0; bnm1 = 1.0;
   x = input("valeur de x: ");
   if abs(x) < 1 then
   xx = x*x;
// cas n = 1; coef de la fraction: x et 1
   an = anm2*x + anm1*1.0;
   bn = bnm2*x + bnm1*1.0;
   mprintf("%12.6f\t%12.6f\t%12.6f\n",an, bn, an/bn);
   anm2 = anm1; bnm2 = bnm1;
   anm1 = an; bnm1 = bn;
//cas général
   n = 2;
   while n <= NMAX
       an = -anm2*xx*(n-1)^2 + anm1*(2*n-1);
       bn = -bnm2*xx*(n-1)^2 + bnm1*(2*n-1);
       anm2 = anm1/bn; bnm2 = bnm1/bn;
    anm1 = an/bn; bnm1 = 1;
       mprintf("%12.6f\t%12.6f\t%12.6f\n",an, bn, an/bn);
       if (abs(anm1/bnm1 - anm2/bnm2) <= TOL), break end
       n = n+1;
   end
   mprintf("après %d itérations, argth(%8.4f) vaut:..
              %15.10f\n", n-1, x, an/bn);
   mprintf("valeur donnée par Scilab: %15.10f\n",atanh(x) );
end
```