Stud. Univ. Babeş-Bolyai Math. 64(2019), No. 2, 291–296
DOI: 10.24193/subbmath.2019.2.13

An application of inverse Padé interpolation

Radu T. Trîmbiţaş

Abstract. We use inverse Padé interpolation to find a fourth order method for the solution of scalar nonlinear equations. Our approach is based on Computer Algebra. Maple Computer Algebra system assisted us to find the method and to establish its order.

Mathematics Subject Classification (2010): 65H05, 65Y99.

Keywords: Nonlinear equations, inverse interpolation, Computer Algebra.

1. Introduction

Suppose we wish to approximate the solution of the nonlinear equation

$$f(x) = 0, \tag{1.1}$$

where $f : \Omega \subseteq \mathbb{R} \to \mathbb{R}$. Let α be a solution of (1.1).

Suppose there exists $g = f^{-1}$ on a neighbourhood V of α. The inverse interpolation consists of approximating

$$\alpha = g(0),$$

by the value of an interpolant $\widehat{g}$ for g at 0

$$\alpha \approx \widehat{g}(0).$$

In this paper we will use inverse Padé interpolation. Let $\mathcal{R}_{m,n}$ be the set of rational functions with numerator degree m and denominator degree n. Suppose f has a formal Taylor series

$$f(z) = c_0 + c_1 z + c_2 z^2 + \cdots$$

For any pair of natural numbers (m, n), $r_{mn} \in \mathcal{R}_{m,n}$ is the type (m, n) Padé approximant to f if their Taylor series at $z = 0$ agree as far as possible:

$$(f - r_{mn})(z) = O\left(z^{\max}\right). \tag{1.2}$$

This paper has been presented at the fourth edition of the International Conference on Numerical Analysis and Approximation Theory (NAAT 2018), Cluj-Napoca, Romania, September 6-9, 2018.

The formula we look for will have the form

$$x_{k+1} = r_{mn}(x_k), \ k = 0, 1, \ldots \tag{1.3}$$

For details on inverse interpolation see [1, 5, 7]. The paper [7] uses rational interpolation to derive methods for the solution of scalar nonlinear equations.

The paper is structured as follows. The second section establishes the formula, its order and the efficiency index. The third section studies the convergence by using a fixed point approach. The next section gives a MATLAB implementation. Finally, the last section gives two numerical examples and compares the new method to Newton and Halley methods respectively.

2. The formula and its order

We use the Maple package `numapprox` (see [4, 6]). Let us start with a $(1, 1)$-degree inverse Padé interpolation.

```
> restart;
> with(numapprox):
> eval(pade((f@@(-1))(y),y=f(x),[1,1]),y=0);
```

$$1/2\,{\frac {4\,x \left({\rm D}\left(f \right) \left(x \right) \right) ^{2}-\left(2\,\left(D^{(2)} \right) \left(f \right) \left(x \right) x+4\,{\rm D}\left(f \right) \left(x \right) \right) f\left(x \right) }{2\,\left({\rm D}\left(f \right) \left(x \right) \right) ^{2}-\left(D^{(2)} \right) \left(f \right) \left(x \right) f\left(x \right) }}$$

```
> collect(%,x,simplify);
```

$$x-2\,{\frac {{\rm D}\left(f \right) \left(x \right) f\left(x \right) }{2\,\left({\rm D}\left(f \right) \left(x \right) \right) ^{2}-\left(D^{(2)} \right) \left(f \right) \left(x \right) f\left(x \right) }}$$

We rewrite the formula as

$$\Phi(x) = x - \frac{f(x)}{f'(x) - \frac{f''(x)f(x)}{2f'(x)}}.$$

This is the well-known Halley's formula. This formula was obtained using direct Padé approximation in [2, 3].

The next step is to try a $(2, 1)$-degree Padé formula.

```
> eval(pade((f@@(-1))(y),y=f(x),[2,1]),y=0):
> F1:=collect(%,x,simplify);
```

$$F1 := x-1/2\,{\frac {f\left(x \right) \left(6\,\left({\rm D}(f)(x) \right) ^{2}\left(D^{(2)} \right) (f)(x)+2\,{\rm D}(f)(x)f(x)\left(D^{(3)} \right) (f)(x)-3\,\left(\left(D^{(2)} \right) (f)(x) \right) ^{2}f(x) \right) }{{\rm D}(f)(x) \left(3\,\left({\rm D}(f)(x) \right) ^{2}\left(D^{(2)} \right) (f)(x)+{\rm D}(f)(x)f(x)\left(D^{(3)} \right) (f)(x)-3\,\left(\left(D^{(2)} \right) (f)(x) \right) ^{2}f(x) \right) }}$$

Rewrite the previous formula as

$$\Phi(x) = x - \frac{f(x)}{f'(x)} \left\{ 1 + \frac{\frac{1}{2}}{\frac{f'(x)}{f''(x)} \left[\frac{f'(x)}{f(x)} + \frac{f'''(x)}{3f''(x)} \right] - 1} \right\}. \tag{2.1}$$

The formula (2.1) is equivalent to `F1` from previous Maple code.

```
> FF:=x-f(x)/D(f)(x)*(1+(1/2)/((D(f)(x)/(D@@2)(f)(x))*(D(f)(x)/f(x)+
> (D@@3)(f)(x)/(3*(D@@2)(f)(x)))-1)):
> simplify(F1-FF);
```

$$0$$

We compute the order of (2.1) as follows:

```
> Phi:=unapply(F1,x);
```

$$\Phi := x \mapsto x - 1/6\,\frac{\left(x^3-a\right)\left(324\,x^5-72\,x^2\left(x^3-a\right)\right)}{x^2\left(162\,x^5-90\,x^2\left(x^3-a\right)\right)}$$

```
> simplify(Phi(alpha),[f(alpha)=0]);
```

$$\alpha$$

```
> simplify(D(Phi)(alpha),[f(alpha)=0]);
```

$$0$$

```
> simplify((D@@2)(Phi)(alpha),[f(alpha)=0]);
```

$$0$$

```
> simplify((D@@3)(Phi)(alpha),[f(alpha)=0]);
```

$$0$$

```
> simplify((D@@4)(Phi)(alpha),[f(alpha)=0]);
```

$$1/3\frac{\left(3\left(D^{(2)}\right)(f)(\alpha)\left(D^{(4)}\right)(f)(\alpha)-4\left(\left(D^{(3)}\right)(f)(\alpha)\right)^2\right)(\mathrm{D}(f)(\alpha))^2-6\mathrm{D}(f)(\alpha)\left(\left(D^{(2)}\right)(f)(\alpha)\right)^2\left(D^{(3)}\right)(f)(\alpha)+9\left(\left(D^{(2)}\right)(f)(\alpha)\right)^4}{(\mathrm{D}(f)(\alpha))^3\left(D^{(2)}\right)(f)(\alpha)}$$

The last expression is the asymptotic error constant. We write it as

$$C_\alpha = \frac{\left(3f''(\alpha)f^{(4)}(\alpha) - 4\left[f'''(\alpha)\right]^2\right)f'^2(\alpha) - 6f'(\alpha)\left[f''(\alpha)\right]^2 f'''(\alpha) + 9\left[f''(\alpha)\right]^4}{3\left[f'(\alpha)\right]^3 f''(\alpha)}. \tag{2.2}$$

The order of (2.1) is $d = 4$, and the efficiency index is $4^{\frac{1}{4}} = \sqrt{2}$. See [5, Section 3.2] for a definition and properties of the efficiency index.

3. The convergence

Let $I_\varepsilon = \{x \in \Omega : |x - \alpha| < \varepsilon\}$ and

$$M(\varepsilon) = \max_{x \in I_\varepsilon} \left| \frac{\Phi^{(4)}(x)}{4!} \right|. \tag{3.1}$$

Theorem 3.1. *If* $\Phi \in C^4(I_\varepsilon)$ *and*

$$\varepsilon^3 M(\varepsilon) < 1, \tag{3.2}$$

then

(a) $x_n \in I_\varepsilon, n = 1, 2, 3 \ldots, \forall x_0 \in I_\varepsilon$;
(b) $\lim_{n\to\infty} x_n = \alpha$.

Proof. (a) Since the method has order $d = 4$ we have

$$e_{n+1} \leq Ce_n^4 \tag{3.3}$$

where $e_n = |x_n - \alpha|$ and M is given by (3.1). If $x_0 \in I_\varepsilon$, the conclusion follows by using complete induction.

(b) From (3.3) using the Maple code

```
> rsolve({e(n+1)=C*e(n)^4,e(0)=e0},e(n));
```

$$\frac{e0^{4^n} C^{1/3 \cdot 4^n}}{\sqrt[3]{C}}$$

it follows

$$e_{n+1} \leq \frac{\left[\left(Ce_0^3\right)^{4^n}\right]^{1/3}}{C^{1/3}}.$$

The right hand side tends to 0 if $Ce_0^3 < 1$ which is equivalent to (3.2). □

[1, Theorem 26.1.4, pag. 317] leads us to the same conclusion.

4. Implementation

The function `invPade` gives a MATLAB implementation for the formula (2.1). The meaning of the input and output parameters are explained in function header.

```
function [y,ni]=invPade(f, fp1, fp2, fp3, x0, ea, er, nmax)
%INVPADE - solution of f(x) = 0 by inverse Pade interpolation
%f, fp1, fp2, fp3 - f and its derivatives
%x0 - starting value
%ea, er - absolute and relative error
%nmax - maximum number of iterations
%y - result
%ni - #iterations

if nargin < 8, nmax=50; end;
if nargin < 7, er=0; end
if nargin < 6, ea=1e-4; end
for k=1:nmax
  f0=f(x0); f1=fp1(x0); f2=fp2(x0); f3=fp3(x0);
  ffp=f0/f1; ifp=f1/f0;
  x1=x0-ffp*(1+0.5/(f1/f2*(ifp+f3/3/f2)-1));
  if abs(x1-x0)<ea+er*abs(x1) %success
    y=x1; ni=k;
    return
  end
  x0=x1;
end
error('max #iterations exceeded')
```

5. Numerical examples

We tested our implementation at the computation of $\sqrt[3]{a}$. We compared our method to Newton and Halley method, respectively. See the source below. We took $a = 201$.

```
a = input('a=');
f = @(x) x^3-a;
fd1 = @(x) 3*x^2;
fd2 = @(x) 6*x;
fd3 = @(x) 6;

[z0,ni0]=Newton(f,fd1,(a+2)/3, 0, eps,100)
[z1,ni1]=Halley(f,fd1,fd2,(a+2)/3, 0, eps, 100)
[z2,ni2]=invPade(f,fd1,fd2,fd3,(a+2)/3, 0, eps, 100)
z0 =
5.857766002650652
ni0 =
12
z1 =
5.857766002650653
ni1 =
8
z2 =
5.857766002650652
ni2 =
6
```

In order to compute the result with a relative error of machine epsilon and the starting value $x_0 = (a+2)/3$, inverse Padé method requires 6 iteration, while Newton and Halley method require 12 and 8 iterations respectively.

The second example solves numerically the equation

$$xe^x + x^2 - 6 = 0.$$

```
g = @(x) x*exp(x)+x^2-6;
gd1 = @(x) (x+1)*exp(x)+2*x;
gd2 = @(x) 2+(x+2)*exp(x);
gd3 = @(x) (x+3)*exp(x);
tic
[z0,ni0]=Newton(g,gd1,5, 0, eps,100);
t0=toc;
tic
[z1,ni1]=Halley(g,gd1,gd2, 5, 0, eps,100);
t1=toc;
tic
```

```
[z2,ni2]=invPade(g,gd1,gd2,gd3, 5, 0, eps,100);
t2=toc;
fprintf('Newton,    z=%17.15f, ni=%2d, elapsed time=%f\n',z0,ni0,t0)
fprintf('Halley,    z=%17.15f, ni=%2d, elapsed time=%f\n',z1,ni1,t1)
fprintf('Inv. Pade, z=%17.15f, ni=%2d, elapsed time=%f\n',z2,ni2,t2)
```

```
Newton,    z=1.257169468081542, ni=11, elapsed time=0.000570
Halley,    z=1.257169468081542, ni= 6, elapsed time=0.000560
Inv. Pade, z=1.257169468081542, ni= 5, elapsed time=0.000549
```

For a relative tolerance equal to machine epsilon and a starting value $x_0 = 5$, the inverse Padé method requires 5 iterations, while Newton and Halley method require 11 and 6 iterations, respectively.

References

[1] Agratini, O., Blaga, P., Chiorean, I., Coman, Gh., Stancu, D.D., Trîmbiţaş, R.T., *Numerical Analysis and Approximation Theory* (vol. III), Cluj University Press, Cluj-Napoca, 2002 (in Romanian).

[2] Gander, W., Gander, M.J., Kwok, F., *Scientific Computing. An Introduction Using Maple and MATLAB*, Springer, 2014.

[3] Gander, W., Gruntz, D., *Derivation of numerical methods using Computer Algebra*, SIAM Rev., **41**(1999), no. 3, 577-593.

[4] Garvan, F., *The Maple Book*, 1st Edition, Chapman & Hall/CRC, 2001.

[5] Gautschi, W., *Numerical Analysis*, Second Edition, Springer Science+Business Media, 2012.

[6] Heck, A., *Introduction to Maple*, Third Edition, Springer-Verlag, New York, 2003.

[7] Păvăloiu, I., *Equations Solution through Interpolation*, Dacia Publishers, 1981 (in Romanian).

Radu T. Trîmbiţaş
Babeş-Bolyai University
Faculty of Mathematics and Computer Sciences
1, Kogălniceanu Street,
400084 Cluj-Napoca, Romania
e-mail: tradu@math.ubbcluj.ro