



**HAL**  
open science

## Padé-type approximants for generalized Euler transforms

Paul Sablonnière

► **To cite this version:**

Paul Sablonnière. Padé-type approximants for generalized Euler transforms. Numerical Algorithms, 2014, 66 (2), pp.339-347. 10.1007/s11075-013-9737-4 . hal-00786065

**HAL Id: hal-00786065**

**<https://hal.science/hal-00786065>**

Submitted on 7 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Padé-type approximants for generalized Euler transforms

P. Sablonnière, INSA & IRMAR, Rennes

February 7, 2013

## Abstract

We prove that sequences generated by the generalized Euler's transform can be considered as Padé-type approximants obtained by Hermite interpolation of the generating function  $u \rightarrow (1 + xu)^{-1}$  at the endpoints of the interval  $[0, 1]$ . A first natural extension is then proposed by considering Hermite interpolation at multiple points of larger intervals.

## 1 Introduction

Consider a formal power series (abbr. fps) defined through a linear functional  $c$  acting on polynomials via the generating function  $g(x, u) = (1 + xu)^{-1}$  as follows

$$f(x) = c((1 + x \cdot)^{-1}) = \sum_{k \geq 0} (-1)^k c_k x^k, \quad c_k = \text{moments} = c(u^k).$$

The *generalized Euler's transform* (abbr. GET, see e.g. [3, 4, 5]) applied to the sequence of partial sums of this series:

$$S_p(x) := \sum_{k=0}^p (-1)^k c_k x^k, \quad p \geq 0$$

consists in computing the triangular array of sequences defined, for  $q \geq 0$ , by:

$$S_p^{(0)} = S_p(x), \quad S_p^{(q+1)}(x) = \frac{x}{1+x} S_p^{(q)}(x) + \frac{1}{1+x} S_{p+1}^{(q)}(x).$$

We prove that sequences generated by the GET can be considered as *Padé type approximants* (abbr. PTAs, see [1]) *obtained by Hermite interpolation of the generating function  $g(x, \cdot)$  at the endpoints of the interval  $[0, 1]$* . A natural extension is then proposed by considering Hermite interpolation at multiple points of a larger interval.

The paper is organised as follows. In Section 2 is proved a recurrence relationship between error estimates on Hermite polynomial interpolants at the endpoints of  $[0, 1]$ . In Section 3, this recurrence relationship is extended to associated PTAs. In Section 4 is proved the main result of the paper, i.e. the connection between PTAs and sequences generated by the GET. Finally, in Section 5, we propose a generalisation that is illustrated with some numerical examples.

## 2 Two-point Hermite interpolation of the function $g(x, \cdot)$

For  $(p, q) \in \mathbb{N} \times \mathbb{N}$ , we use the following notations for multiple points,

$$X_{p,q} := \{0^p, 1^q\} = \{0 \text{ (} p \text{ times), } 1 \text{ (} q \text{ times)}\}$$

$[X_{p,q}]f$  = divided difference of the function  $f$  on the set  $X_{p,q}$ .

The two-point Hermite interpolating polynomial  $P_{p,q}(x, u)$  (of degree  $n = p + q - 1$  in the variable  $u$ ) of the function  $g(x, u)$  satisfies the following  $p + q$  boundary conditions ( $D = d/du$ ):

$$D^r P_{p,q}(x, 0) = D^r g(x, 0) = (-1)^r r! \quad 0 \leq r \leq p - 1,$$

$$D^s P_{p,q}(x, 1) = D^s g(x, 1) = (-1)^s s! (1 + x)^{-(s+1)}, \quad 0 \leq s \leq q - 1.$$

The interpolation error can be written as (see e.g. [2], chapter III):

$$E_{p,q}(x, u) := g(x, u) - P_{p,q}(x, u) = u^p(u - 1)^q [X_{p,q}^u] g(x, \cdot)$$

where  $[X_{p,q}^u]g(x, \cdot)$  denotes the divided difference of order  $n + 1$  based on the set of points  $X_{p,q}^u := X_{p,q} \cup \{u\}$  of the function  $g$  with respect to its second variable.

**Theorem.** *The following recurrence relation holds for interpolation errors:*

$$E_{p,q+1}(x, u) = \frac{x}{1+x} E_{p,q}(x, u) + \frac{1}{1+x} E_{p+1,q}(x, u)$$

This leads to the general formula

$$E_{p,q}(x, u) = (-1)^{n+1} \frac{x^{n+1} u^p (u - 1)^q}{(1+x)^q (1+xu)}$$

*Proof.* For the sake of simplicity, we use the abridged notation  $[X_{p,q}]g$  for the divided differences of the function  $g(x, \cdot)$  on the set  $X_{p,q}$ . For the first values of  $(p, q)$ , the theorem is satisfied since one has

$$E_{1,1}(x, u) = u(u - 1)[0, u, 1]g$$

$$E_{2,1}(x, u) = u^2(u - 1)[0, 0, u, 1]g, \quad E_{1,2}(x, u) = u(u - 1)^2[0, u, 1, 1]g$$

and, by direct computation of the first finite differences:

$$[0, u, 1]g = \frac{x^2}{(1+x)(1+xu)}$$

$$[0, 0, u]g = \frac{x^2}{1+xu}, \quad [u, 1, 1]g = \frac{x^2}{(1+x)^2(1+xu)}$$

Moreover, we have

$$[0, 0, u, 1]g = [0, u, 1]g - [0, 0, u]g = \frac{-x^3}{(1+x)(1+xu)}$$

$$[0, u, 1, 1]g = [u, 1, 1]g - [0, u, 1]g = \frac{-x^3}{(1+x)^2(1+xu)}$$

Then, we first obtain

$$\begin{aligned} \frac{x}{1+x}E_{1,1}(x, u) + \frac{1}{1+x}E_{2,1}(x, u) &= \frac{x}{1+x}u(u-1)[0, u, 1]g + \frac{1}{1+x}u^2(u-1)[0, 0, u, 1]g \\ &= \frac{x^3u(u-1)}{(1+x)^2(1+xu)} - \frac{x^3u^2(u-1)}{(1+x)^2(1+xu)} = \frac{-x^3u(u-1)^2}{(1+x)^2} \end{aligned}$$

On the other hand, we get

$$E_{1,2}(x, u) = u(u-1)^2[0, u, 1, 1]g = \frac{-x^3u(u-1)^2}{(1+x)^2(1+xu)}$$

i.e. the same expression.

Assume now that for  $p+q = n-1$ , we have already proved

$$E_{p,q}(x, u) = u^p(u-1)^q[X_{p,q}^u]g = (-1)^{n+1} \frac{x^{n+1}u^p(u-1)^q}{(1+x)^q(1+xu)}.$$

(This is true for  $(p, q) = (1, 1), (2, 1), (1, 2)$  from the above computation).

Then on the one hand, we have

$$E_{p+1,q}(x, u) = u^{p+1}(u-1)^q[X_{p+1,q}^u]g$$

with

$$[X_{p+1,q}^u]g = [X_{p,q}^u]g - [X_{p+1,q-1}^u]g = \frac{(-1)^{n+1}x^{n+1}}{(1+x)^q(1+xu)} - \frac{(-1)^{n+1}x^{n+1}}{(1+x)^{q-1}(1+xu)} = \frac{(-1)^{n+2}x^{n+2}}{(1+x)^q(1+xu)}.$$

On the other hand, we have

$$E_{p,q+1}(x, u) = u^p(u-1)^{q+1}[X_{p,q+1}^u]g$$

with

$$[X_{p,q+1}^u]g = [X_{p-1,q+1}^u]g - [X_{p,q}^u]g = \frac{(-1)^{n+1}x^{n+1}}{(1+x)^{q+1}(1+xu)} - \frac{(-1)^{n+1}x^{n+1}}{(1+x)^q(1+xu)} = \frac{(-1)^{n+2}x^{n+2}}{(1+x)^{q+1}(1+xu)}.$$

Finally, by comparison of the above expressions of  $E_{p,q}(x, u)$ ,  $E_{p+1,q}(x, u)$  and  $E_{p,q+1}(x, u)$ , it is easy to verify that the following recursion holds

$$E_{p,q+1}(x, u) = \frac{x}{1+x}E_{p,q}(x, u) + \frac{1}{1+x}E_{p+1,q}(x, u).$$

The general formula follows immediately by induction. ■

The interpolant being given by

$$P_{p,q}(x, u) := g(x, u) - E_{p,q}(x, u) = \frac{1}{1+xu} \left( 1 - (-1)^{n+1} \frac{x^{n+1}u^p(u-1)^q}{(1+x)^q} \right)$$

it is straightforward to deduce the following

**Theorem.** *The Hermite interpolants of  $g(x, \cdot)$  satisfy the recurrence relationship*

$$P_{p,q+1}(x, u) = \frac{x}{1+x}P_{p,q}(x, u) + \frac{1}{1+x}P_{p+1,q}(x, u)$$

### 3 Padé-type approximants and generalised Euler transform

In this section, we show the connection between the preceding Hermite interpolating polynomials, the associated PTAs and the rational functions generated by the GET from the partial sums of the series  $f(x) = \sum_{k \geq 0} (-1)^k c_k x^k = c((1+x)^{-1})$ .

Following Brezinski ([1], chapter I), we define the PTAs of  $f$  as follows

$$[p, q]f(x) := c(P_{p,q}(x, \cdot))$$

From the preceding theorem, we deduce

**Theorem.** *The PTAs of  $f$  satisfy the recurrence relationship*

$$[p, q+1]f(x) = \frac{x}{1+x}[p, q]f(x) + \frac{1}{1+x}[p+1, q]f(x)$$

The starting values are, for  $n = p + q - 1 = 1$  or  $2$ :

$$[1/1]_f(x) = c(P_{11}(x, \cdot)) = c\left(\frac{1}{1+xu} - \frac{x^2}{1+x} \frac{u(u-1)}{1+xu}\right)$$

$$[2/1]_f(x) = c(P_{21}(x, \cdot)) = c\left(\frac{1}{1+xu} + \frac{x^3}{1+x} \frac{u^2(u-1)}{1+xu}\right)$$

$$[1/2]_f(x) = c(P_{12}(x, \cdot)) = c\left(\frac{1}{1+xu} + \frac{x^3}{(1+x)^2} \frac{u(u-1)^2}{1+xu}\right)$$

Moreover, we have

$$c\left(\frac{u(u-1)}{1+xu}\right) = c\left(\frac{u^2}{1+xu}\right) - c\left(\frac{u}{1+xu}\right) = \sum_{k \geq 0} (-1)^k \Delta c_{k+1} x^k$$

and, in a similar way

$$c\left(\frac{u^2(u-1)}{1+xu}\right) = c\left(\frac{u^3}{1+xu}\right) - c\left(\frac{u^2}{1+xu}\right) = \sum_{k \geq 0} (-1)^k \Delta c_{k+2} x^k$$

$$c\left(\frac{u(u-1)^2}{1+xu}\right) = c\left(\frac{(u^3 - u^2)}{1+xu}\right) - c\left(\frac{(u^2 - u)}{1+xu}\right) = \sum_{k \geq 0} (-1)^k \Delta^2 c_{k+1} x^k$$

Therefore we obtain

$$[1/1]_f(x) = f(x) - \frac{x}{1+x} \sum_{k \geq 1} (-1)^{k-1} \Delta c_k x^k$$

$$[1/1]_f(x) = \sum_{k \geq 0} (-1)^k c_k x^k - \frac{1}{1+x} \sum_{k \geq 2} (-1)^k (c_k - c_{k-1}) x^k$$

$$\begin{aligned}
&= \frac{1}{1+x} \left( \sum_{k \geq 0} (-1)^k c_k x^k - \sum_{k \geq 2} (-1)^k c_k x^k \right) + \frac{x}{1+x} \left( \sum_{k \geq 0} (-1)^k c_k x^k + \sum_{k \geq 2} (-1)^k c_{k-1} x^{k-1} \right) \\
&= \frac{1}{1+x} (c_0 - c_1 x) + \frac{x}{1+x} (c_0) = \frac{x}{1+x} S_0(x) + \frac{1}{1+x} S_1(x)
\end{aligned}$$

Therefore we get the first step of GET:

$$[1/1]_f(x) = S_0^{(1)}(x)$$

Similar computations lead to

$$[2/1]_f(x) = S_1^{(1)}(x) \quad \text{and} \quad [1/2]_f(x) = S_0^{(2)}(x)$$

For the general proof, we first observe that the Hermite interpolants at  $u = 0$  of the function  $g(x, \cdot)$  are the Maclaurin expansions

$$P_{p,0}(x, u) = \sum_{k=0}^{p-1} (-1)^k u^k x^k$$

Therefore, by definition of PTAs,

$$[p, 0]_f(x) = c(P_{p,0}(x, \cdot)) = \sum_{k=0}^{p-1} (-1)^k c_k x^k = S_{p-1}^{(0)}(x), \quad \forall p \geq 1.$$

Assume that we have already proved that for some  $q \geq 1$ :

$$[p, q]_f(x) = S_{p-1}^{(q)}(x) \quad \forall p \geq 1.$$

Then we have, by definition,

$$S_{p-1}^{(q+1)}(x) = \frac{x}{1+x} S_{p-1}^{(q)}(x) + \frac{1}{1+x} S_p^{(q)}(x) \quad \forall p \geq 1,$$

and we also know that the PTAs satisfy

$$[p, q+1]_f(x) = \frac{x}{1+x} [p, q]_f(x) + \frac{1}{1+x} [p+1, q]_f(x) \quad \forall p \geq 1.$$

Therefore we deduce immediately

$$S_{p-1}^{(q+1)}(x) = [p, q+1]_f(x), \quad \forall p \geq 1, \quad \text{q.e.d.} \quad \blacksquare$$

## 4 A new extension of the Euler transform

The above results can be extended in various ways. As an example, we show that, starting from Hermite interpolants of  $g(x, \cdot)$  at the three points  $u = 0, 1, 2$ , and computing the associated PTAs, it is possible to obtain a new extension of the generalised Euler transform.

As in Section 2, we use the following notations:

$$X_{p,q,r} := \{0^{(p)}, 1^{(q)}, 2^{(r)}\} = \{0 \text{ (} p \text{ times), } 1 \text{ (} q \text{ times), } 2 \text{ (} r \text{ times)}\}$$

$$[X_{p,q,r}^u]g := [0^{(p)}, u, 1^{(q)}, 2^{(r)}]g(x, \cdot) = [0^{(p)}, 1^{(q)}, u, 2^{(r)}]g(x, \cdot)$$

respectively for the set of data points (for Hermite interpolation) and the divided differences of the generating function  $g(x, \cdot) : u \rightarrow (1+xu)^{-1}$ . The interpolating polynomial (of degree  $n = p + q + r - 1$ ) and the associated interpolation error are then

$$P_{p,q,r}(x, u), \quad E_{p,q,r}(x, u) := g(x, u) - P_{p,q,r}(x, u) = u^p(u-1)^q(u-2)^r[X_{p,q,r}^u]g(x, \cdot)$$

#### 4.1 Recurrence relations

A general expression of the error is given by

$$E_{p,q,r}(x, u) = \frac{(-x)^{n+1}}{(1+x)^q(1+2x)^r} \frac{u^p(u-1)^q(u-2)^r}{1+xu}.$$

As in Section 2, it is obtained in a straightforward manner by induction on indices  $p, q, r$ . From that, one deduces the following recurrence relationships

$$E_{p,q+1,r}(x, u) = \frac{x}{1+x}E_{p,q,r}(x, u) + \frac{1}{1+x}E_{p+1,q,r}(x, u)$$

$$E_{p,q,r+1}(x, u) = \frac{x}{1+2x}E_{p,q,r}(x, u) + \frac{1+x}{1+2x}E_{p,q+1,r}(x, u)$$

$$E_{p,q,r+1}(x, u) = \frac{2x}{1+2x}E_{p,q,r}(x, u) + \frac{1}{1+2x}E_{p+1,q,r}(x, u)$$

Now, introducing the (generalised) PTAs

$$[p/q/r]f(x) := c(P_{p,q,r}(x, \cdot))$$

one can define an extension of the generalised Euler transform (GET) as follows.

As we have seen in Section 3, the partial sums are  $S_p(x) = S_p^{(0)}(x) = [p+1/0]f(x)$ , and the classical GET was defined by  $S_p^{(q)}(x) = [p+1, q]f(x)$ . They both constitute the level  $r = 0$  of the new GET, so we slightly modify the notations and we now set, instead of the preceding notation,

$$s_{p,q}^{[0]}(x) = [p+1, q]f(x)$$

More generally, for all  $r \geq 0$ , we define the array at the level  $r$  by

$$s_{p,q}^{[r]}(x) = [p+1, q, r]f(x) \quad \forall p, q \geq 0$$

## 4.2 Algorithm for the extended GET

Here is an algorithm for the computation of the sequences generated by the extended GET.

- Step 1 (Level  $r = 0$ ): computation of the partial sums  $s_{p,0}^{(0)} = S_p(x), 0 \leq p \leq n$ .
- Step 2 (Level  $r = 0$ ): computation of the sums  $s_{p,q}^{(0)}(x), 0 \leq p \leq n - q$  for  $q = 1, \dots, n$  by the recurrence relationship of the classical GET

$$s_{p,q+1}^{(0)}(x) = \frac{x}{1+x} s_{p,q}^{(0)}(x) + \frac{1}{1+x} s_{p+1,q}^{(0)}(x).$$

- Step 3 (Levels  $r \geq 1$ ). For each level  $r = 1, \dots, n$ , compute the array  $s_{p,q}^{[r]}(x)$  by using one of the recurrence relations deduced from those of the preceding subsection:

$$s_{p,q}^{[r+1]}(x) = \frac{2x}{1+2x} s_{p,q}^{[r]}(x) + \frac{1}{1+2x} s_{p+1,q}^{[r]}(x)$$

or

$$s_{p,q}^{[r+1]}(x) = \frac{x}{1+2x} s_{p,q}^{[r]}(x) + \frac{1+x}{1+2x} s_{p,q+1}^{[r]}(x)$$

for  $q = 0, \dots, n - r$  and  $0 \leq p \leq n - r - q$ .

## 5 Some numerical experiments

### 5.1 Power series

We consider some of the power series studied in [5] and we show the interest of using GET of higher levels for large values of  $x > 0$ .

**Example (P1):**  $f(x) = \ln(1+x)/x$  for  $x = 6, 8, 10, 12$ , which gives strongly divergent Maclaurin series. Table 1 shows the absolute values of best errors obtained by GET at each level  $r = 0, 1, 2$ , using  $n = 20$  terms of the series.

$x$	6	8	10	12
$r = 0$	4.3(-4)	1.3(-3)	2.6(-3)	3.9(-3)
$r = 1$	2.7(-5)	1.5(-4)	4.1(-4)	8.2(-4)
$r = 2$	2.9(-5)	8.9(-6)	6.7(-5)	2.0(-4)

**Example (P2):**  $f(x) = (x - x^2/2 + x^3/3 - \ln(1+x))/x^4$  for  $x = 1, 2, 5, 6$ , which also gives strongly divergent series. Table 2 below shows the absolute values of the best errors obtained by GET at each level  $r = 0, 1, 2$ , obtained with  $n = 20$  terms of the series.

$x$	1	2	5	6
$r = 0$	1.5(-10)	3.0(-7)	2.4(-4)	6.1(-3)
$r = 1$	3.1(-12)	3.0(-9)	1.0(-5)	4.7(-5)
$r = 2$	4.8(-9)	5.1(-7)	2.7(-6)	1.7(-5)

In both examples, one can notice that, for large values of  $x > 0$ , higher levels of GET can give better results than the classical GET.



## 5.2 Nilakantha and Euler series

$$(N1) \quad \pi = 2\sqrt{3} \sum_{k \geq 1} \frac{(-1)^{k-1}}{(2k-1)3^{k-1}}, \quad (N2) \quad \pi = 16 \sum_{k \geq 1} \frac{(-1)^{k-1}}{(2k-1)^5 + 4 \cdot (2k-1)}$$

$$(E1) \quad \frac{\pi}{4} = \sum_{k \geq 1} \frac{(-1)^{k-1}}{(2k-1)},$$

$$(E2) \quad \frac{\pi^3}{32} = \sum_{k \geq 1} \frac{(-1)^{k-1}}{(2k-1)^3}, \quad (E3) \quad \frac{7\pi^4}{720} = \sum_{k \geq 1} \frac{(-1)^{k-1}}{k^4}$$

The table below shows the absolute values of best errors obtained by GET at each level  $r = 0, 1, 2, 3$ , obtained with  $n = 20$  terms of the series for  $x = 1$ .

$x$	N1	N2	E1	E2	E3
$r = 0$	1.8(-11)	8.3(-12)	1.5(-10)	2.1(-11)	3.5(-11)
$r = 1$	8.2(-14)	3.3(-14)	3.1(-12)	2.8(-14)	2.0(-14)
$r = 2$	3.0(-17)	3.5(-14)	4.8(-9)	4.8(-12)	1.0(-12)
$r = 3$	1.0(-17)	2.9(-12)	2.1(-7)	2.7(-10)	1.1(-10)

Again, for these series, one can observe that the results obtained with the first levels of the extended GET can be much better than those obtained with the classical GET.

It would be interesting to make a deeper study of the preceding method and also to test other types of Hermite interpolants of the generating function in order to eventually obtain better summation methods.

## References

- [1] C. Brezinski, *Padé type approximants and generalized orthogonal polynomials* ISNM Vol. **50**, Birkhäuser-Verlag, Basel, 1980.
- [2] P.J. Davis, *Interpolation and approximation*, reprint, Dover, 1975.
- [3] S.A. Gustafson, Convergence acceleration on a general class of power series. *Computing* **21** (1978), 53-69.
- [4] I.M. Longman, A note on the summation of power series. *Appl. Numer. Math.* **4** (1988), 431-437.
- [5] P. Sablonnière, A short note on the generalized Euler transform for the summation of power series, *Numer. Algor.* **2** (1992), 241-254.