Applied Numerical Mathematics 6 (1989/90) 169–185
North-Holland

OLD AND NEW MULTIDIMENSIONAL CONVERGENCE ACCELERATORS

Annie CUYT
Department of Mathematics and Computer Science, University of Antwerp, Universiteitsplein 1, B-2610 Wilrijk, Belgium

In the past some multidimensional convergence accelerators have been studied by Levin [13], by Albertsen, Jacobsen and Sørensen [1] and by the author [5]. We show here that all these multidimensional convergence accelerators are particular cases of a whole class of multidimensional convergence accelerators. The common underlying principle is that they can be considered as multivariate Padé approximants for a multivariate function that is different for different algorithms. Since we work in a very general framework, we are able to introduce a number of new multidimensional convergence accelerators and generalize them by using multivariate rational Hermite interpolants instead of multivariate Padé approximants.

1. Convergence acceleration of a table with single entry

The idea of using the epsilon-algorithm to accelerate the convergence of a sequence, which can be considered as a table with single entry, is quite well-known. Given the sequence $\{a_i\}_{i\in\mathbb{N}}$ with $A=\lim_{i\to\infty}a_i$, we choose n and m in $\mathbb{N}$ and construct the ratio of determinants

$$\frac{\begin{vmatrix} a_n & a_{n-1} & \cdots & a_{n-m} \\ \nabla a_{n+1} & \nabla a_n & \cdots & \nabla a_{n+1-m} \\ \vdots & \vdots & \ddots & \vdots \\ \nabla a_{n+m} & \nabla a_{n+m-1} & \cdots & \nabla a_n \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & & 1 \\ \nabla a_{n+1} & \nabla a_n & \cdots & \nabla a_{n+1-m} \\ \vdots & \vdots & \ddots & \vdots \\ \nabla a_{n+m} & \nabla a_{n+m-1} & \cdots & \nabla a_n \end{vmatrix}} \tag{1}$$

with $\nabla a_i=a_i-a_{i-1}$ and $a_i=0$ for $i<0$. The ratio (1) is the Padé approximant of order $(n,\ m)$ evaluated at $x=1$ for the univariate function

$$f(x)=\sum_{i=0}^{\infty}\nabla a_i x^i.$$

We are particularly interested in approximations at $x=1$ since

$$f(1)=A.$$

This ratio of determinants can easily be computed using the epsilon-algorithm. With

$$\varepsilon_{-1}^{(k)} = 0, \quad \varepsilon_0^{(k)} = a_k, \qquad k = 0, 1, \ldots,$$
$$\varepsilon_{k+1}^{(l)} = \varepsilon_{k-1}^{(l+1)} + \frac{1}{\varepsilon_k^{(l+1)} - \varepsilon_k^{(l)}}, \qquad k = 0, 1, \ldots, \quad l = 0, 1, \ldots,$$

formula (1) is given by $\varepsilon_{2m}^{(n-m)}$. The ε-values are usually arranged in a table as follows

$$\begin{array}{ccccc}
\varepsilon_{-1}^{(0)} & & & & \\
 & \varepsilon_0^{(0)} & & & \\
\varepsilon_{-1}^{(1)} & & \varepsilon_1^{(0)} & & \\
 & \varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & \\
\varepsilon_{-1}^{(2)} & & \varepsilon_1^{(1)} & & \ddots \\
\vdots & \varepsilon_0^{(2)} & \vdots & \varepsilon_2^{(1)} & \\
 & \vdots & & \vdots & \ddots
\end{array}$$

The epsilon-algorithm is called a convergence accelerator because it can be shown, in some cases, that the convergence of the columns or diagonals in the ε-table is faster than that of the given sequence $\{a_i\}_{i \in \mathbb{N}}$ [2, pp. 83–85].

The previous reasoning was generalized by Brezinski. Given sequences $\{g_k(i)\}_{i \in \mathbb{N}}$, $k = 1, 2, \ldots$, and $\{a_i\}_{i \in \mathbb{N}}$ with $A = \lim_{i \to \infty} a_i$ where approximately

$$a_i = A + \alpha_1 g_1(i) + \cdots + \alpha_m g_m(i), \quad i \geqslant 0,$$

it is easy to see that an approximate value for A is given by

$$\frac{\begin{vmatrix} a_n & a_{n-1} & \cdots & a_{n-m} \\ g_1(n) & & \cdots & g_1(n-m) \\ \vdots & & & \vdots \\ g_m(n) & & \cdots & g_m(n-m) \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ g_1(n) & \cdots & g_1(n-m) \\ \vdots & & \vdots \\ g_m(n) & \cdots & g_m(n-m) \end{vmatrix}} \tag{2}$$

In [3] it is shown that formula (2) can be computed recursively in an analogous way as formula (1), now using the E-algorithm. With

$$E_0^{(l)} = a_l, \quad l = 0, 1, \ldots, n,$$
$$g_{0,k}^{(l)} = g_k(l), \qquad k = 1, 2, \ldots, m, \quad l = 0, 1, \ldots, n,$$
$$E_{k+1}^{(l)} = \frac{E_k^{(l)} g_{k,k+1}^{(l+1)} - E_k^{(l+1)} g_{k,k+1}^{(l)}}{g_{k,k+1}^{(l+1)} - g_{k,k+1}^{(l)}}, \qquad l = 0, 1, \ldots, \quad k = 0, 1, \ldots,$$
$$g_{k+1,j}^{(l)} = \frac{g_{k,j}^{(l)} g_{k,k+1}^{(l+1)} - g_{k,j}^{(l+1)} g_{k,k+1}^{(l)}}{g_{k,k+1}^{(l+1)} - g_{k,k+1}^{(l)}}, \qquad j = k+2, k+3, \ldots, \quad k = 0, 1, \ldots,$$

formula (2) is given by $E_m^{(n-m)}$. The values $E_{k+1}^{(l)}$ and $g_{k+1,j}^{(l)}$ are stored as in the tables (B.1) and (B.2) of Appendix B. Convergence acceleration results are given in [3]. In [10] a computationally more advantageous organization of the E-algorithm is given in order to compute formula (2). With

$$E_{0,0}^{(l)} = a_l/g_1(l), \quad E_{0,1}^{(l)} = 1/g_1(l), \qquad l = 0, 1, \ldots, n,$$

$$h_{0,k}^{(l)} = g_k(l)/g_1(l), \qquad k = 2, \ldots, m, \quad l = 0, 1, \ldots, n,$$

$$h_{k,k+1}^{(l)} = 1, \quad k = 0, 1, \ldots,$$

$$E_{k,p}^{(l)} = \frac{E_{k-1,p}^{(l+1)} - E_{k-1,p}^{(l)}}{h_{k-1,k+1}^{(l+1)} - h_{k-1,k+1}^{(l)}}, \qquad l = 0, 1, \ldots, \quad k = 1, 2, \ldots, \quad p = 0, 1,$$

$$h_{k,j}^{(l)} = \frac{h_{k-1,j}^{(l+1)} - h_{k-1,j}^{(l)}}{h_{k-1,k+1}^{(l+1)} - h_{k-1,k+1}^{(l)}}, \qquad j = k+2, k+3, \ldots, \quad k = 1, 2, \ldots,$$

formula (2) is given by $E_{m,0}^{(n-m)}/E_{m,1}^{(n-m)}$. The values $E_{k,0}^{(l)}$, $E_{k,1}^{(l)}$ and $h_{k,j}^{(l)}$ are stored as in the tables (B.3) and (B.4) of Appendix B. So far for the one-dimensional case.

2. Convergence acceleration of a table with multiple entry

Suppose we are given a table $\{a_{i_1 \ldots i_p}\}_{(i_1,\ldots,i_p) \in \mathbb{N}^p}$ with multiple entry and with $A = \lim_{i_1,\ldots,i_p \to \infty} a_{i_1 \ldots i_p}$. The convergence accelerators we shall propose are different from the ones suggested by Haccart in [12]. She extracts a one-dimensional subsequence from the multidimensional table and applies one-dimensional convergence accelerators. We preserve the multidimensional nature of the problem as is done in [1,13]. In [5] formula (1) is generalized for this case as follows. Define

$$f(x_1, \ldots, x_p) = \sum_{i_1, \ldots, i_p = 0}^{\infty} {}_p\nabla a_{i_1 \ldots i_p} x_1^{i_1} \ldots x_p^{i_p} \tag{3a}$$

with

$$\begin{aligned} {}_p\nabla a_{i_1 \ldots i_p} = a_{i_1 \ldots i_p} &- \sum_{j=1}^{p} a_{i_1 \ldots (i_j - 1) \ldots i_p} \\ &+ \sum_{\substack{j,k=1 \\ j<k}}^{p} a_{i_1 \ldots i_{j-1}(i_j-1)i_{j+1} \ldots i_{k-1}(i_k-1)i_{k+1} \ldots i_p} - \cdots + (-1)^p a_{(i_1-1)\ldots(i_p-1)}. \end{aligned} \tag{3b}$$

Clearly

$$f(1, \ldots, 1) = A.$$

For this multivariate function multivariate Padé approximants can be calculated and evaluated at the point $(x_1, \ldots, x_p) = (1, \ldots, 1)$ via the epsilon-algorithm [5]. Let us restrict ourselves to the bivariate case for the sake of notational simplicity and deal with a table $\{a_{ij}\}_{(i,j) \in \mathbb{N}}$ of double entry. The bivariate Padé approximant of order (n, m) to which the epsilon-algorithm applies, is then given by

$$\frac{\begin{vmatrix} \sum_{i+j=0}^{n} {}_2\nabla a_{ij} & \cdots & \sum_{i+j=0}^{n-m} {}_2\nabla a_{ij} \\ \sum_{i+j=n+1} {}_2\nabla a_{ij} & \cdots & \sum_{i+j=n+1-m} {}_2\nabla a_{ij} \\ \vdots & & \vdots \\ \sum_{i+j=n+m} {}_2\nabla a_{ij} & \cdots & \sum_{i+j=n} {}_2\nabla a_{ij} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \sum_{i+j=n+1} {}_2\nabla a_{ij} & \cdots & \sum_{i+j=n+1-m} {}_2\nabla a_{ij} \\ \vdots & & \vdots \\ \sum_{i+j=n+m} {}_2\nabla a_{ij} & \cdots & \sum_{i+j=n} {}_2\nabla a_{ij} \end{vmatrix}}, \tag{4}$$

where $\Sigma_{i+j=0}^{k}\, {}_2\nabla a_{ij}$ can be simplified to $\Sigma_{i+j=k} a_{ij} - \Sigma_{i+j=k-1} a_{ij}$. For convergence acceleration results we refer to [5].

Just as the one-dimensional formula (1) can be generalized to the multidimensional case by using (3), the one-dimensional expression (2) can be translated to the multidimensional case. Then its recursive computation scheme will again be based on the E-algorithm. We shall now show that a number of multidimensional convergence accelerators suggested in the past appear to be particular cases of this generalization and also that some new multidimensional convergence accelerators can be introduced. The common underlying principle is that they can be considered as multivariate Padé approximants either for a multivariate function $f(x_1, \ldots, x_p)$ different from the one used in (3a) or with different multivariate numerator and denominator than in (4). For a description of the definition of multivariate Padé approximant that is used here, we refer to [8]. The framework developed there is a very general one and covers many previously introduced definitions for multivariate Padé approximants. The reader that is not familiar with [8] can first consult Appendix A.

As a first special case, consider a convergence accelerator analogous to the ones developed by Levin in [13] and by Albertsen, Jacobsen and Sørensen in [1], given by

$$\frac{\begin{vmatrix} a_{i_n-d_0, j_n-e_0} & \cdots & a_{i_n-d_m, j_n-e_m} \\ \nabla_0 a_{i_{n+1}-d_0, j_{n+1}-e_0} & \cdots & \nabla_m a_{i_{n+1}-d_m, j_{n+1}-e_m} \\ \vdots & & \vdots \\ \nabla_0 a_{i_{n+m}-d_0, j_{n+m}-e_0} & \cdots & \nabla_m a_{i_{n+m}-d_m, j_{n+m}-e_m} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \nabla_0 a_{i_{n+1}-d_0, j_{n+1}-e_0} & \cdots & \nabla_m a_{i_{n+1}-d_m, j_{n+1}-e_m} \\ \vdots & & \vdots \\ \nabla_0 a_{i_{n+m}-d_0, j_{n+m}-e_0} & \cdots & \nabla_m a_{i_{n+m}-d_m, j_{n+m}-e_m} \end{vmatrix}}, \tag{5}$$

where

$$\nabla_k a_{i_h-d_k,j_h-e_k} = a_{i_h-d_k,j_h-e_k} - a_{i_{h-1}-d_k,j_{h-1}-e_k}, \qquad a_{i_n-d_k,j_n-e_k} = \sum_{h=0}^{n} \nabla_k a_{i_h-d_k,j_h-e_k}.$$

Remember that $(i_0, j_0), (i_1, j_1), (i_1, j_2), \ldots$ is an enumeration of $\mathbb{N}^2$. That this expression can be computed by means of the E-algorithm can easily be seen from the fact that it is a multivariate Padé aproximant as given in [8] for the function

$$f(x, y) = \sum_{h=0}^{\infty} \nabla a_{i_h j_h} x^{i_h} y^{j_h}, \tag{6}$$

where $\nabla a_{i_h j_h} = a_{i_h j_h} - a_{i_{h-1} j_{h-1}}$ and $a_{ij} = 0$ if $i < 0$ or $j < 0$. In the table of E-values expression (5) can be found as $E_m^{(n)}$ or as $E_{m,0}^{(n)}/E_{m,1}^{(n)}$. When Levin and Albertsen, Jacobsen and Sørensen developed their convergence accelerator, they did not know that recursive computation was possible and they computed (5) by solving systems of linear equations. The starting values for the E-algorithm are given by

$$E_0^{(l)} = \sum_{(i_h,\, j_h) \in N_l} \nabla a_{i_h j_h} x^{i_h} y^{j_h} \Big|_{(x,y)=(1,1)} = \sum_{h=0}^{l} \nabla a_{i_h j_h} = a_{i_l j_l}$$

$$g_{0,r}^{(l)} = \sum_{(i_h,\, j_h) \in N_l} \nabla_r a_{i_h-d_r,j_h-e_r} x^{i_h-d_r} y^{i_h-e_r} \Big|_{(x,y)=(1,1)}$$

$$= \sum_{h=0}^{l} \nabla_r a_{i_h-d_r,j_h-e_r} = a_{i_l-d_r,j_l-e_r}$$

and for its simplified form by

$$E_{0,0}^{(l)} = a_{i_l j_l}/a_{i_l-d_1,j_l-e_1}, \qquad E_{0,1}^{(l)} = 1/a_{i_l-d_1,j_l-e_1}, \qquad h_{0,r-1}^{(l)} = a_{i_l-d_r,j_l-e_r}.$$

In the table of E-values expression (5) can be found as $E_m^{(n)}$ or as $E_{m,0}^{(n)}/E_{m,1}^{(n)}$. If one examines [8] carefully, it is clear that a whole bunch of convergence accelerators, based on the use of multivariate general order Padé approximants, can be constructed. An even more general formula than (5) is the one that results if we use multivariate rational interpolants instead of multivariate Padé approximants. We refer the reader to the following section.

3. Some new multidimensional convergence accelerators

Let us first discuss new methods that result from the use of multivariate general order Padé approximants. Consider a two-dimensional table $\{a_{ij}\}_{(i,j)\in\mathbb{N}^2}$ with $\lim_{i,j\to\infty} a_{ij} = A$ and construct the series (3) with $f(1, 1) = A$. If we choose a numbering in $\mathbb{N}^2$ and construct a sequence of sets $\{I_l\}_{l\in\mathbb{N}}$ with $I_l = \{(i_0, j_0), (i_1, j_1), \ldots, (i_l, j_l)\}$, then for each $l = 0, 1, \ldots$ a variety of numerator and denominator index sets N_n and D_m with $n + m = l$ exists. The general order Padé approximants $[N_n/D_m]_{I_l}(1, 1)$ for (3), as described in [8] and summarized in Appendix A, can be computed by means of the E-algorithm and are found in $E_m^{(n)}$ or in $E_{m,0}^{(n)}/E_{m,1}^{(n)}$. Its starting values are also given in Appendix A. A determinant expression for $[N_n/D_m]_{I_l}(1, 1)$ is given by

$$\frac{\begin{vmatrix} \sum_{h=0}^{n} {}_2\nabla_0 a_{i_h, j_h} & \cdots & \sum_{h=0}^{n} {}_2\nabla_m a_{i_h, j_h} \\ {}_2\nabla_0 a_{i_{n+1}, j_{n+1}} & \cdots & {}_2\nabla_m a_{i_{n+1}, j_{n+1}} \\ \vdots & & \vdots \\ {}_2\nabla_0 a_{i_{n+m}, j_{n+m}} & \cdots & {}_2\nabla_m a_{i_{n+m}, j_{n+m}} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ {}_2\nabla_0 a_{i_{n+1}, j_{n+1}} & \cdots & {}_2\nabla_m a_{i_{n+1}, j_{n+1}} \\ \vdots & & \vdots \\ {}_2\nabla_0 a_{i_{n+m}, j_{n+m}} & \cdots & {}_2\nabla_m a_{i_{n+m}, j_{n+m}} \end{vmatrix}} \tag{7}$$

with

$${}_2\nabla_k a_{i_h, j_h} = a_{i_h - d_k, j_h - e_k} - a_{i_h - d_k - 1, j_h - e_k} - a_{i_h - d_k, j_h - e_k - 1} + a_{i_h - d_k - 1, j_h - e_k - 1}.$$

In order to illustrate this technique numerically we consider the following situation. Suppose one wants to calculate the integral of a function $u(x, y)$ on a bounded closed domain Ω of $\mathbb{R}^2$. For the sake of simplicity we take $\Omega = [0, 1] \times [0, 1]$. The table $\{a_{ij}\}_{(i,j) \in \mathbb{N}^2}$ can for instance be obtained by subdividing the interval [0, 1] in each direction respectively into 2^i and 2^j intervals of equal length $h_1 = 2^{-i}$ and $h_2 = 2^{-j}$. Using the midpoint rule one can then substitute approximations

$$\int_0^{h_1} \int_0^{h_2} u(x, y) \, dx \, dy = h_1 h_2 u\left(\tfrac{1}{2}h_1, \tfrac{1}{2}h_2\right)$$

to calculate

$$a_{ij} = \frac{1}{2^{i+j}} \left(\sum_{k=1}^{2^i} \sum_{l=1}^{2^j} u\left(\frac{2k-1}{2^{i+1}}, \frac{2l-1}{2^{j+1}} \right) \right).$$

Let us take the diagonal enumeration of $\mathbb{N}^2$ given by (0, 0), (1, 0), (0, 1), (2, 0), (1, 1), (0, 2),.... As an example we take $u(x, y) = 1/(x + y)$ which produces slowly converging a_{ij} because of the singularity of the integrand in (0, 0). Let the values a_{ij} be given for $0 \leqslant i + j \leqslant 9$. With these data the approximations (4), (5) and (7) can be computed respectively for $n + m = 0, \ldots, 9$ and $n + m = 0, \ldots, 54$. Note that (4) adds a complete diagonal of data in one step and (5) and (7) add the data along a diagonal one by one. The results displayed in Table 1 are the most accurate among the possible approximations $\varepsilon_{2m}^{(n-m)}$ and $E_m^{(n)}$ respectively for $n + m = i$ and $n + m + 1 = \frac{1}{2}(i + 1)(i + 2)$ where $i \leqslant 9$. We can compare them with the $a_{\lfloor (i+1)/2 \rfloor, \lfloor i/2 \rfloor}$. The exact value of the integral is

$$\int_0^1 \int_0^1 \frac{1}{x + y} \, dx \, dy = 2 \ln 2 = 1.38629436\ldots.$$

It is clear that (5) can be improved by (7). Similar conclusions can be found in the sequal of this section.

Table 1

$a_{\lfloor (i+1)/2\rfloor, \lfloor i/2\rfloor}$	$\varepsilon_{2\lfloor i/2\rfloor}^{\lfloor (i+1)/2\rfloor - \lfloor i/2\rfloor}$	Formula (7)	Formula (5)
$a_{11} = 1.166667$	$\varepsilon_2^{(0)} = 1.330295$	$E_2^{(3)} = 1.292352$	$E_2^{(3)} = 1.183518$
$a_{21} = 1.209102$	$\varepsilon_2^{(1)} = 1.361764$	$E_3^{(6)} = 1.374224$	$E_3^{(6)} = 1.228489$
$a_{22} = 1.269048$	$\varepsilon_4^{(0)} = 1.396396$	$E_5^{(9)} = 1.359011$	$E_5^{(9)} = 1.304007$
$a_{32} = 1.292978$	$\varepsilon_4^{(1)} = 1.386057$	$E_8^{(12)} = 1.373649$	$E_8^{(12)} = 1.329994$
$a_{33} = 1.325744$	$\varepsilon_6^{(0)} = 1.386872$	$E_{10}^{(17)} = 1.385863$	$E_{10}^{(17)} = 1.360150$
$a_{43} = 1.338426$	$\varepsilon_6^{(1)} = 1.386481$	$E_{15}^{(20)} = 1.386177$	$E_{15}^{(20)} = 1.374274$
$a_{44} = 1.355532$	$\varepsilon_8^{(0)} = 1.386309$	$E_{18}^{(26)} = 1.386366$	$E_{18}^{(26)} = 1.371675$
$a_{54} = 1.362056$	$\varepsilon_8^{(1)} = 1.386298$	$E_{25}^{(29)} = 1.386298$	$E_{25}^{(29)} = 1.385897$

Secondly we shall discuss a technique that results from the use of multivariate general order rational interpolants described in [8] and summarized in Appendix A. Consider a two-dimensional table $\{a_{ij}\}_{(i,j)\in\mathbb{N}^2}$ with $\lim_{i,j\to\infty} a_{ij} = A$ and let $\{(x_i, y_j)\}_{(i,j)\in\mathbb{N}^2}$ be a convergent table of points in $\mathbb{R}^2$ with

$$\lim_{i,j\to\infty} (x_i, y_j) = (z_1, z_2).$$

When using extrapolation techniques to accelerate the convergence of $\{a_{ij}\}_{(i,j)\in\mathbb{N}^2}$, We compute a sequence $\{b_i\}_{i\in\mathbb{N}}$ with

$$b_i = \lim_{(x,y)\to(z_1,z_2)} s_i(x, y),$$

where $s_i(x, y)$ is determined by some interpolation conditions. The point (z_1, z_2) is called the extrapolation point. In analogy with the univariate extrapolation technique of Bulirsch-Stoer, we choose for $s_i(x, y)$ the bivariate rational interpolants on the descending staircase

$$\begin{array}{lll}
[N_0/D_0]_{I_0} & & \\
[N_1/D_0]_{I_1} & [N_1/D_1]_{I_2} & \\
 & [N_2/D_1]_{I_3} & [N_2/D_2]_{I_4} \\
 & \vdots & \ddots
\end{array}$$

These rational interpolants,

$$[N_n/D_m]_{I_{n+m}} = \frac{p_n(x, y)}{q_m(x, y)},$$

are constructed here such that

$$a_{ij} = \frac{p_n(x_i, y_j)}{q_m(x_i, y_j)}, \quad (i, j) \in I_{n+m} \subseteq \mathbb{N}^2,$$

and then b_i is computed from

$$b_i = \frac{p_n(z_1, z_2)}{q_m(z_1, z_2)}, \qquad n = \lfloor \tfrac{1}{2}(i+1) \rfloor, \quad m = \lfloor \tfrac{1}{2} i \rfloor, \quad i = 0, 1, 2, \ldots. \tag{8}$$

Of course the choice of the interpolation points (x_i, y_j) greatly influences the convergence behaviour of the resulting sequence $\{b_i\}_{i \in \mathbb{N}}$.

Let us compare the formulas (5), (7) and (8) numerically. We know that the Beta function $B(x, y)$ for $-1 < x, y < 0$ can be written as

$$B(x, y) = \frac{1 + xyw(x+1, y+1)}{xy(x+1)(y+1)}(x+y)(1+x+y),$$

and that a Taylor series development for $w(x+1, y+1)$ can be computed by the first method suggested in [11]. Let us denote this Taylor series by

$$w(x+1, y+1) = \sum_{i, j=0}^{\infty} c_{0i,0j} x^i y^j,$$

and its partial sums for $(x, y) = (u, v)$ by

$$a_{kl} = \sum_{i=0}^{k} \sum_{j=0}^{l} c_{0i,0j} u^i v^j.$$

Input of the convergence accelerators are these bivariate partial sums of the Taylor series development for $w(x+1, y+1)$ around (1, 1). We shall compute rational approximants and interpolants $R(u, v)$ for $w(u+1, v+1)$ and compare the value

$$\frac{1 + uvR(u, v)}{uv(u+1)(v+1)}(u+v)(1+u+v)$$

with $B(u, v)$. Let the values a_{kl} be given for $(k, l) \in I \subseteq \mathbb{N}^2$. Let us associate each a_{kl} with an interpolation point (x_k, y_l) where

$$\lim_{k,l \to \infty} (x_k, y_l) = (z_1, z_2), \qquad \lim_{k,l \to \infty} a_{kl} = w(u, v).$$

So we construct a function $f(x, y)$ satisfying

$$f(x_k, y_l) = a_{kl}, \qquad f(z_1, z_2) = w(u, v).$$

We can then proceed as in the Padé approximation case, choosing numerator and denominator index sets N and D, constructing subsets N_n, D_m and I_l, computing rational interpolants $[N_n/D_m]_{I_{n+m}}$ for $f(x, y)$ and evaluating them at (z_1, z_2). In our example we have taken $(u, v) = (-0.92, -0.97)$.

In the first column of Table 2 the values,

$$\frac{1 + uva_{ii}}{uv(u+1)(v+1)}(u+v)(1+u+v),$$

are displayed. For the construction of all the other columns we have taken

$$I = \{(k, l): 0 \leqslant k, l \leqslant 6\},$$

and used the enumeration (0, 0), (1, 0), (0, 1), (1, 1), (2, 0), (0, 2), (2, 1), (1, 2), (2, 2), (3, 0), (0, 3), (3, 1), (1, 3), (3, 2), (2, 3), (3, 3),.... In order to compare values that use the same number of data as the a_{ii} for $i = 0, 1, 2, \ldots$, we choose for $R(x, y)$ the $(i+1)^2$th elements on the descending

Table 2

i	a_{ii}	(5)	(7)	(8a)	(8b)
1	118.551	106.835	140.066	23.0915	23.0915
2	92.1841	88.6788	83.1745	100.396	88.2749
3	88.3833	87.6991	86.8761	78.0438	85.7630
4	87.1083	73.6345	86.1894	86.5946	86.0138
5	86.5533	93.2420	86.0873	83.8083	86.0543
6	86.3002	85.8025	86.0793	86.8151	86.0689

staircase given at the beginning of this section, namely

$$\begin{aligned} i=0:&\quad [N_0/D_0]_{I_0},\\ i=1:&\quad [N_2/D_1]_{I_3},\\ i=2:&\quad [N_4/D_4]_{I_8},\\ i=3:&\quad [N_8/D_7]_{I_{15}},\\ i=4:&\quad [N_{12}/D_{12}]_{I_{24}},\\ i=5:&\quad [N_{18}/D_{17}]_{I_{35}}.\end{aligned}$$

The second column is an illustration of (5) and the third column an illustration of (7). The other columns illustrate (8) where we have made different choices for the interpolation points:

$$(x_k,\ y_l)=\left(\frac{1}{k+1},\frac{1}{l+1}\right), \tag{8a}$$

$$(x_k,\ y_l)=(2^{-k},2^{-l}). \tag{8b}$$

The correct limit is

$$A=B(-0.92,\ -0.97)=86.07672\ldots.$$

All computations are performed on a Gould UTX/32 in double precision arithmetic. Remark that the values a_{ii} converge slowly due to the presence of singularities in the vincinity of $(u, v)=(-0.92,\ -0.97)$, namely in $u=-1$ and $v=-1$.

When evaluating the multidimensional convergence accelerators overall we advice to consider construction of the series (3) if Padé approximants are used and to pay attention to the choice of the interpolation points if rational interpolants are used.

4. Exact summation

Several theorems exist that describe the type of series which can be summed exactly by a particular convergence accelerator in that sense that an application of the convergence accelerator to the sequence of its partial sums gives the limit value A. Consider for instance formula (1) again. According to [2, pp. 40–42] a necessary and sufficient condition for

$$\varepsilon_{2m}^{(n-m)}=A,\quad n-m=l,\ l+1,\ldots,$$

is that there exist constants $\alpha_0, \ldots, \alpha_m$ with $\Sigma_{k=0}^m \alpha_k \neq 0$ and $\alpha_0 \alpha_m \neq 0$, such that

$$\sum_{k=0}^{m} \alpha_k (a_{n-m+k} - A) = 0, \quad n-m = l,\ l+1, \ldots. \tag{9}$$

This can easily be seen by solving (9) using Cramer's rule.

When using (2) instead of (1), formula (9) generalizes as follows. A necessary and sufficient condition for the exact summation [3]

$$E_m^{(n-m)} = A, \quad n-m = l,\ l+1, \ldots,$$

is

$$a_{n-m} = A + \sum_{k=1}^{m} \alpha_k g_k(n-m), \quad n-m = l,\ l+1, \ldots. \tag{10}$$

When turning to the multivariate case analogous conclusions can be written down. The summation process (4) sums the series

$$\sum_{i,\, j=0}^{\infty} \nabla a_{ij}$$

exactly, if

$$\sum_{k=0}^{m} \alpha_k \left(\sum_{i+j=0}^{n-m+k} \nabla a_{ij} - A \right) = 0 \tag{11}$$

with $\Sigma_{k=0}^m \alpha_k \neq 0$ and $\alpha_0 \alpha_m \neq 0$.

An even more general result was proved in [4] for (5). When computed recursively, expression (5) is given by $E_m^{(n)}$. A necessary and sufficient condition for exact summation of

$$\sum_{h=0}^{\infty} \nabla a_{i_h j_h} \tag{12}$$

is that there exist $\alpha_0, \ldots, \alpha_m$ with $\alpha_0 \alpha_m \neq 0$ and not all zero such that

$$\sum_{k=0}^{m} \alpha_k \left(a_{i_n - d_k,\, j_n - e_k} - A \right) = 0.$$

We have not mentioned the most general results here because this would take away much of the clarity when we try to show how things generalize. The reader can now translate more general conditions than (9) and (10) to the multivariate case. We conclude with a necessary and sufficient condition for our new convergence accelerator to sum the series (12) exactly. It is a simple application of results given in [4].

Corollary. *A necessary and sufficient condition for exact summation of the series* (12) *by* $[N_n/D_m]_{I_{n+m}} = (p_n/q_m)(x, y)$ $(n \geq l)$, *satisfying*

$$\frac{p_n}{q_m}(x_i, y_j) = a_{ij}, \quad (i, j) \in I_{n+m},$$

is that there exist $\alpha_0, \ldots, \alpha_m$ *not all zero with* $\alpha_0\alpha_m \neq 0$ *and such that for* $n \geqslant l$

$$\alpha_0(t_0(n) - A) + \cdots + \alpha_m(t_m(n) - A) = 0,$$

where, in the notation of Appendix A, for $0 \leqslant k \leqslant m$

$$t_k(0) = c_{d_k i_0, e_k j_0} B_{d_k i_0, e_k j_0}(x, y), \qquad t_k(n) = \sum_{h=0}^{n} c_{d_k i_h, e_k j_h} B_{d_k i_h, e_k j_h}(x, y)$$

with

$$c_{d_k i, e_k j} = f[x_{d_k}, \ldots, x_i][y_{e_k}, \ldots, y_j], \quad f[x_i][y_j] = a_{ij},$$

$$c_{d_k i, e_k j} = 0, \qquad i < d_k \quad or \quad j < e_k.$$

Appendix A. General order Padé approximants and rational Hermite interpolants for multivariate functions

Let us restrict everything to the case of two variables for the sake of simplicity. Furthermore we assume that the finite interpolation set $I = \{(i, j)\colon f_{ij} \text{ is given at } (x_i, y_j)\}$ is structured so that it satisfies the inclusion property. This means that if a point belongs to the data set, then the rectangular subset of points emanating from the origin with the given point as its furthermost corner also lies in the data set. How this can be achieved in a lot of situations is explained in [9]. If none of the points in $\{(x_i, y_j)\}_{(i,j)\in I}$ coincide, then we are dealing with a rational interpolation problem and the values in $\{f_{ij}\}_{(i,j)\in I}$ are function values. If all the interpolation points coincide, then the problem is one of Padé approximation and it is well known that the given data aren't function values but Taylor coefficients. If some of the points coincide and some don't, then the problem is of a mixed type and it is called a Hermite interpolation problem or a Newton–Padé approximation problem. In [9] is indicated how one should interpret the data f_{ij}: some of them are derivatives and some of them are function values. In the sequel of the text it doesn't play a role whether one is dealing with coalescent points or not since all the formulas remain valid in both cases and in the mixed case. Nevertheless we shall sometimes indicate how the formulas are to be read if some of the interpolation points coincide.

Consider the following set of basis functions for the real-valued polynomials in two variables

$$B_{ij}(x, y) = \prod_{k=0}^{i-1} (x - x_k) \prod_{l=0}^{j-1} (y - y_l).$$

Clearly $B_{ij}(x, y)$ is a bivariate polynomial of degree $i + j$. Given the f_{ij}, we can write in a purely formal manner

$$f(x, y) = \sum_{(i, j)\in\mathbb{N}^2} f_{0i,0j} B_{ij}(x, y),$$

where the $f_{0i,0j}$ are the bivariate divided differences

$$f_{0i,0j} = f[x_0, \ldots, x_i][y_0, \ldots, y_j]$$

given by

$$f[x_0,\ldots,x_i][y_0,\ldots,y_j]=\frac{f[x_1,\ldots,x_i][y_0,\ldots,y_j]-f[x_0,\ldots,x_{i-1}][y_0,\ldots,y_j]}{x_i-x_0}$$

or

$$f[x_0,\ldots,x_i][y_0,\ldots,y_j]=\frac{f[x_0,\ldots,x_i][y_1,\ldots,y_j]-f[x_0,\ldots,x_i][y_0,\ldots,y_{j-1}]}{y_j-y_0}$$

with

$$f[x_i][y_j]=f_{ij}.$$

Divided differences with coalescent points $x_i,\ldots,x_{i+k}$ and $y_j,\ldots,y_{j+l}$ are given by

$$f[x_i,\ldots,x_{i+k}][y_j]=\frac{1}{k!}\left.\frac{\partial^k f}{\partial x^k}\right|_{(x_i,y_j)}$$

and

$$f[x_i][y_j,\ldots,y_{j+l}]=\frac{1}{l!}\left.\frac{\partial^l f}{\partial y^l}\right|_{(x_i,y_j)}.$$

In the Padé approximation case, for instance when all the interpolation points $(x_i,\ y_j)$ coincide with (0, 0), the data are Taylor coefficients and we can write in a purely formal manner

$$f(x,\ y)=\sum_{(i,\ j)\in\mathbb{N}^2} f_{0i,0j}x^iy^j$$

with

$$f_{0i,0j}=\frac{1}{i!}\frac{1}{j!}\left.\frac{\partial^{i+j}f}{\partial x^i\,\partial y^j}\right|_{(x,y)=(0,0)}$$

Note that the basis functions $B_{ij}(x,\ y)$ now take the very simple form x^iy^j.

In order to construct rational interpolants or Padé approximants for the given set I we choose two finite index sets N, a subset of I, and D, a subset of $\mathbb{N}^2$, which determine the "degree" of the numerator and denominator and we put as in [9]

$$\begin{aligned}
p(x,\ y)&=\sum_{(i,j)\in N} a_{ij}B_{ij}(x,\ y) && (N \text{ from "numerator"}),\\
q(x,\ y)&=\sum_{(i,j)\in D} b_{ij}B_{ij}(x,\ y) && (D \text{ from "denominator"}),\\
(fq-p)(x,\ y)&=\sum_{(i,j)\in\mathbb{N}^2\setminus I} c_{ij}B_{ij}(x,\ y) && (I \text{ from "interpolation conditions"}).
\end{aligned}\tag{A.1}$$

The rational interpolant $(p/q)(x,\ y)$ will then be denoted by

$$[N/D]_I.$$

Let us introduce a numbering $r(i,\ j)$ of the points in $\mathbb{N}^2$, for example based on the enumeration

$$(0,0), \underbrace{(1,0),(0,1),}_{\text{first diagonal}}\ \underbrace{(2,0),(1,1),(0,2),}_{\text{second diagonal}}\ \underbrace{(3,0),(2,1),(1,2),(0,3),}_{\text{third diagonal}}\ldots.$$

In this case

$$r(i,\ j)=\tfrac{1}{2}(i+j)(i+j+1)+j-i,$$

but other enumerations can be used as well. The only limitation is that the enumeration must be such that for every l, the subset of $\mathbb{N}^2$ containing the first l points satisfies the inclusion property too. If we denote $\#N=n+1$ then we can write

$$N=\bigcup_{l=0}^{n} N_l$$

with

$$\begin{aligned}&\emptyset=N_{-1}\subset N_0\subset N_1\subset\cdots\subset N_{n-1}\subset N_n=N,\\ &\#N_l=l+1,\\ &N_l\backslash N_{l-1}=\{(i_l,\ j_l)\},\quad l=0,\ldots,n,\\ &r(i_l,\ j_l)>r(i_r,\ j_r),\quad l>r.\end{aligned}$$

In other words, for each $l=0,\ldots,n$ we add to N_{l-1} the point $(i_l,\ j_l)$ which is the next in line in $N\cap\mathbb{N}^2$ according to the enumeration given above. Denote $\#D=m+1$ and proceed in the same way. Then

$$D=\bigcup_{l=0}^{m} D_l$$

with

$$\begin{aligned}&D_{-1}=\emptyset,\\ &\#D_l=l+1,\quad D_l\backslash D_{l-1}=\{(d_l,\ e_l)\},\qquad l=0,\ldots,m.\end{aligned}$$

Now (A.1) can be rewritten as

$$\begin{aligned}&(fq)_{0i,0j}=p_{0i,0j}=a_{ij}=\sum_{\mu=0}^{i}\sum_{\nu=0}^{j}b_{\mu\nu}f_{\mu i,\nu j},\quad (i,\ j)\in N,\\ &(fq)_{0i,0j}=0=\sum_{\mu=0}^{i}\sum_{\nu=0}^{j}b_{\mu\nu}f_{\mu i,\nu j},\quad (i,\ j)\in I\backslash N,\end{aligned}\tag{A.2}$$

with $b_{\mu,\nu}=0$ for $(\mu,\ \nu)\notin D$. In the Padé approximation case the divided difference $f_{\mu i,\nu j}$ equals the Taylor coefficient $f_{0i-\mu,0j-\nu}$.

Let us assume that the interpolation set $I\backslash N$ is such that exactly m of the homogeneous equations (A.2) are linearly independent. Degenerate cases can be avoided by adding interpolation data to the set I until the rank of (A.2) is equal to m. It is obvious that this condition guarantees the existence of a nontrivial solution of (A.2) given by the following determinant expressions, because the number of unknowns in the homogeneous system is now one more than its rank. We group the respective m elements of $I \backslash N$ that supply the linearly independent equations in the set H and number them also following the enumeration given above,

$$H=\bigcup_{l=1}^{m} H_l \subseteq I \backslash N$$

with

$$H_0=\emptyset$$
$$\# H_l=l, \quad H_l \backslash H_{l-1}=\{(h_l, k_l)\}, \qquad l=1, \ldots, m .$$

The polynomials $p(x, y)$ and $q(x, y)$ satisfying (A.1) are then given by [9]

$$p(x, y)=\begin{vmatrix} \Sigma_{(i, j) \in N} f_{d_0 i, e_0 j} B_{ij}(x, y) & \ldots & \Sigma_{(i, j) \in N} f_{d_m i, e_m j} B_{ij}(x, y) \\ f_{d_0 h_1, e_0 k_1} & \ldots & f_{d_m h_1, e_m k_1} \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} & \ldots & f_{d_m h_m, e_m k_m} \end{vmatrix}, \tag{A.3a}$$

$$q(x, y)=\begin{vmatrix} B_{d_0 e_0}(x, y) & \ldots & B_{d_m e_m}(x, y) \\ f_{d_0 h_1, e_0 k_1} & \ldots & f_{d_m h_1, e_m k_1} \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} & \ldots & f_{d_m h_m, e_m k_m} \end{vmatrix}, \tag{A.3b}$$

where

$$f_{d_i h_j, e_i k_j}=f\left[x_{d_i}, \ldots, x_{h_j}\right]\left[y_{e_i}, \ldots, y_{k_j}\right]$$

with

$$f_{d_i h_j, e_i k_j}=0 \qquad \text{if } d_i>h_j \quad \text{or} \quad e_i>k_j .$$

In [8] these determinant formulas are given when all the interpolation points coincide and a lot of specific choices for N, D and I are described. In [7] it is illustrated that the covariance properties satisfied by these multivariate Padé approximants are determined by the structure of the index sets N, D and I.

The formulas (A.3) can be rewritten so that they can be computed recursively. Multiplying the $(l+1)$th row in $p(x, y)$ and $q(x, y)$ by $B_{h_l k_l}(x, y)$, $l=1, \ldots, m$, and dividing the $(l+1)$th column by $B_{d_l e_l}(x, y)$, $l=0, \ldots, m$, results in

$$p(x, y)=\begin{vmatrix} \Sigma_{(i, j) \in N} f_{d_0 i, e_0 j} B_{d_0 i, e_0 j}(x, y) & \ldots & \Sigma_{(i, j) \in N} f_{d_m i, e_m j} B_{d_m i, e_m j}(x, y) \\ f_{d_0 h_1, e_0 k_1} B_{d_0 h_1, e_0 k_1}(x, y) & \ldots & f_{d_m h_1, e_m k_1} B_{d_m h_1, e_m k_1}(x, y) \\ \vdots & & \vdots \\ f_{d_0 h_m, e_0 k_m} B_{d_0 h_m, e_0 k_m}(x, y) & \ldots & f_{d_m h_m, e_m k_m} B_{d_m h_m, e_m k_m}(x, y) \end{vmatrix}, \tag{A.4a}$$

$$q(x, y) = \begin{vmatrix} 1 & \cdots & 1 \\ f_{d_0h_1,e_0k_1}B_{d_0h_1,e_0k_1}(x, y) & \cdots & f_{d_mh_1,e_mk_1}B_{d_mh_1,e_mk_1}(x, y) \\ \vdots & & \vdots \\ f_{d_0h_m,e_0k_m}B_{d_0h_m,e_0k_m}(x, y) & \cdots & f_{d_mh_m,e_mk_m}B_{d_mh_m,e_mk_m}(x, y) \end{vmatrix}, \tag{A.4b}$$

where for $k \leqslant i$ and $l \leqslant j$

$$B_{ki,lj}(x, y) = \frac{B_{ij}(x, y)}{B_{kl}(x, y)} = (x - x_k) \cdots (x - x_{i-1})(y - y_l) \cdots (y - y_{j-1})$$

and for $k > i$ or $l > j$

$$f_{ki,lj} = 0$$

For such a quotient of determinants the E-algorithm is particularly suitable [6]

$$E_0^{(l)} = \sum_{(i,j) \in N_l} f_{d_0i,e_0j}B_{d_0i,e_0j}(x, y), \quad l = 0, \ldots, n + m,$$

$$g_{0,r}^{(l)} = \sum_{(i,j) \in N_l} f_{d_ri,e_rj}B_{d_ri,e_rj}(x, y) - \sum_{(i,\ j) \in N_l} f_{d_{r-1}i,e_{r-1}j}B_{d_{r-1}i,e_{r-1}j}(x, y),$$

$$r = 1, \ldots, m, \quad l = 0, \ldots, n + m,$$

$$E_r^{(l)} = \frac{E_{r-1}^{(l)}g_{r-1,r}^{(l+1)} - E_{r-1}^{(l+1)}g_{r-1,r}^{(l)}}{g_{r-1,r}^{(l+1)} - g_{r-1,r}^{(l)}}, \quad l = 0, 1, \ldots, n, \quad r = 1, 2, \ldots, m, \tag{A.5a}$$

$$g_{r,s}^{(l)} = \frac{g_{r-1,s}^{(l)}g_{r-1,r}^{(l+1)} - g_{r-1,s}^{(l+1)}g_{r-1,r}^{(l)}}{g_{r-1,r}^{(l+1)} - g_{r-1,r}^{(l)}}, \quad s = r + 1, \ r + 2, \ldots. \tag{A.5b}$$

As a result of these computations

$$[N/D]_I = E_m^{(n)}.$$

Since the solution $p(x, y)/q(x, y)$ of (A.2) is unique due to the fact that the rank of (A.2) is m, the value $E_m^{(n)}$ itself does not depend upon the numbering of the points within the sets N, D and H. But this numbering affects the interpolation conditions satisfied by the intermediate E-values. For $l = 0, \ldots, n$ and $r = 0, \ldots, m$ we prove in [6] that

$$E_r^{(l)} = [N_l/D_r]_{N_l \cup \underbrace{\{(i_{l+1},\ j_{l+1}), \ldots, (i_n,\ j_n), (h_1,\ k_1), \ldots, (h_{r-n+l},\ k_{r-n+l})\}}_{r \text{ points}}}.$$

Appendix B. Tables

The values $E_r^{(l)}$ and $g_{r,s}^{(l)}$ are stored as indicated below.

$$
\begin{array}{ccccc}
E_0^{(0)} & & & & \\
 & E_1^{(0)} & & & \\
E_0^{(1)} & & \ddots & & \\
 & E_1^{(1)} & E_m^{(0)} & & \\
E_0^{(2)} & \vdots & \vdots & \ddots & \\
\vdots & & & & E_{n+m}^{(0)} \\
 & & E_m^{(n)} & & \\
 & E_1^{(n+m-1)} & & & \\
E_0^{(n+m)} & & & &
\end{array}
\tag{B.1}
$$

$$
\begin{array}{c|cc|c|ccc|c|cc}
g_{0,1}^{(0)} & g_{0,2}^{(0)} & \ddots & & g_{0,r}^{(0)} & & & & g_{0,m}^{(0)} & \\
 & & g_{1,2}^{(0)} & & & g_{1,r}^{(0)} & & & & \ddots \\
g_{0,1}^{(1)} & g_{0,2}^{(1)} & & & g_{0,r}^{(1)} & & \ddots & & g_{0,m}^{(1)} & \\
 & & g_{1,2}^{(1)} & & & g_{1,r}^{(1)} & g_{r-1,r}^{(0)} & & \vdots & g_{m-1,m}^{(0)} \\
g_{0,1}^{(2)} & g_{0,2}^{(2)} & \vdots & \cdots & g_{0,r}^{(2)} & \vdots & \vdots & \cdots & & \vdots \\
\vdots & \vdots & & & \vdots & & g_{r-1,r}^{(n+m-r+1)} & & & g_{m-1,m}^{(n+1)} \\
 & & g_{1,2}^{(n+m-1)} & & & g_{1,r}^{(n+m-1)} \; {\cdot}^{\cdot^{\cdot}} & & & & {\cdot}^{\cdot^{\cdot}} \\
g_{0,1}^{(n+m)} & g_{0,2}^{(n+m)} & & & g_{0,r}^{(n+m)} & & & & g_{0,m}^{(n+m)} &
\end{array}
\tag{B.2}
$$

The values $E_{r,p}^{(l)}$ for $p = 0$, 1 and the values $h_{r,s-1}^{(l)}$ are stored as indicated below.

$$
\begin{array}{ccccc}
E_{0,0}^{(0)} & & & & \\
 & E_{1,0}^{(0)} & & & \\
E_{0,0}^{(1)} & & \ddots & & \\
 & E_{1,0}^{(1)} & E_{m,0}^{(0)} & & \\
E_{0,0}^{(2)} & \vdots & \vdots & \ddots & \\
\vdots & & & & E_{n+m,0}^{(0)} \\
 & & E_{m,0}^{(n)} & & \\
 & E_{1,0}^{(n+m-1)} & & & \\
E_{0,0}^{(n+m)} & & & &
\end{array}
\tag{B.3a}
$$

$$
\begin{array}{ccccc}
E_{0,1}^{(0)} & & & & \\
 & E_{1,1}^{(0)} & & & \\
E_{0,1}^{(1)} & & \ddots & & \\
 & E_{1,1}^{(1)} & E_{m,1}^{(0)} & & \\
E_{0,1}^{(2)} & \vdots & \vdots & \ddots & \\
\vdots & & & & E_{n+m,1}^{(0)} \\
 & & E_{m,1}^{(n)} & & \\
 & E_{1,1}^{(n+m-1)} & & & \\
E_{0,1}^{(n+m)} & & & &
\end{array}
\tag{B.3b}
$$

$$
\begin{array}{c|cc|c|cccc|c|ccc}
h^{(0)}_{0,2} & h^{(0)}_{0,3} & & & h^{(0)}_{0,r-1} & & & & & h^{(0)}_{0,m+1} & & \\
 & & h^{(0)}_{1,3} & & & h^{(0)}_{1,r-1} & & & & & \ddots & \\
h^{(1)}_{0,2} & h^{(1)}_{0,3} & & & h^{(1)}_{0,r-1} & & \ddots & & & h^{(1)}_{0,m+1} & & \\
 & & h^{(1)}_{1,3} & & & h^{(1)}_{1,r-1} & & h^{(0)}_{r-1,r+1} & & \vdots & & h^{(0)}_{m-1,m+1} \\
h^{(2)}_{0,2} & h^{(2)}_{0,3} & \vdots & \cdots & h^{(2)}_{0,r-1} & \vdots & & \vdots & \cdots & & & \vdots \\
\vdots & \vdots & & & \vdots & & & h^{(n+m-r+1)}_{r-1,r+1} & & & & h^{(n+1)}_{m-1,m+1} \\
 & & h^{(n+m-1)}_{1,3} & & & h^{(n+m-1)}_{1,r-1} & \cdot^{\cdot^{\cdot}} & & & & \cdot^{\cdot^{\cdot}} & \\
h^{(n+m)}_{0,2} & h^{(n+m)}_{0,3} & & & h^{(n+m)}_{0,r-1} & & & & & h^{(n+m)}_{0,m+1} & &
\end{array}
\tag{B.4}
$$

References

[1] N. Albertsen, G. Jacobsen and S. Sørensen, Nonlinear transformations for accelerating the convergence of M-dimensional series, *Math. Comp.* 41 (1983) 623–634.

[2] C. Brezinski, *Accélération de la Convergence en Analyse Numérique*, Lecture Notes in Mathematics 584 (Springer, Berlin, 1977).

[3] C. Brezinski, A general extrapolation algorithm, *Numer. Math.* 35 (1980) 175–187.

[4] C. Brezinski, Composite sequence transformations, *Numer. Math.* 46 (1985) 311–321.

[5] A. Cuyt, Accelerating the convergence of a table with multiple entry, *Numer. Math.* 41 (1983) 281–286.

[6] A. Cuyt, A recursive computation scheme for multivariate rational interpolants, *SIAM J. Numer. Anal.* 24 (1987) 228–238.

[7] A. Cuyt, A review of multivariate Padé approximation theory, *J. Comput. Appl. Math.* 12–13 (1985) 221–232.

[8] A. Cuyt, Multivariate Padé approximants revisited, *BIT* 26 (1986) 71–79.

[9] A. Cuyt and B. Verdonk, General order Newton-Padé approximants for multivariate functions, *Numer. Math.* 43 (1984) 293–307.

[10] W. Ford and A. Sidi, An algorithm for a generalization of the Richardson extrapolation process, *SIAM J. Numer. Anal.* 24 (1987) 1212–1232.

[11] P. Graves-Morris, R. Hughes Jones and G. Makinson, The calculation of some rational approximants in two variables, *J. Inst. Math. Appl.* 13 (1974) 311–320.

[12] I. Haccart, Accélération de la convergence de suites, séries et intégrales doubles, Thèse 3ème cycle, Université de Lille (1983).

[13] D. Levin, On accelerating the convergence of infinite double series and integrals, *Math. Comp.* 35 (1980) 1331–1346.