
From numerical quadrature to Padé approximation

Claude Brezinski
University of Lille - France

- Numerical quadrature
 - Interpolatory quadratures
 - Gaussian quadratures
 - Error estimates
- Padé approximation of power series
 - Padé-type approximants
 - Padé approximants
 - Error estimates
 - The ε -algorithm
- Perspectives

The definition of Padé approximants is well known

$$f(t) = \sum_{i=0}^{\infty} c_i t^i$$

$$[p/q]_f(t) = P(t)/Q(t)$$

$$f(t)Q(t) - P(t) = \mathcal{O}(t^{p+q+1})$$

I will show that Padé approximants can also be derived from numerical analysis methods.

NUMERICAL QUADRATURE

Consider the definite integral

$$I(g) = \int_a^b g(x)w(x) dx,$$

where w is a weight function satisfying $\forall x \in]a, b[, w(x) > 0$, integrable in $[a, b]$, and where a and b can be finite or infinite.

The mapping $I : g \longmapsto I(g)$ is linear, and it maps a space of functions into the set of real numbers. Thus, it is a **linear functional**, also called a **linear form**.

INTERPOLATORY QUADRATURES

An approximate value of $I(g)$ can be obtained by replacing g by its interpolating polynomial R_{k-1} , and computing $I(R_{k-1})$.

Such a procedure is called an

interpolatory quadrature method.

The polynomial R_{k-1} is given by Lagrange's formula

$$R_{k-1}(x) = \sum_{i=1}^k L_i^{(k-1)}(x)g(x_i),$$

with, for $i = 1, \dots, k$,

$$L_i^{(k-1)}(x) = \prod_{\substack{j=1 \\ j \neq i}}^k \frac{x - x_j}{x_i - x_j} = \frac{v_k(x)}{(x - x_i)v_k'(x_i)}, \quad v_k(x) = \prod_{i=1}^k (x - x_i).$$

Thus

$$I(R_{k-1}) = \sum_{i=1}^k A_i^{(k-1)} g(x_i),$$

with

$$A_i^{(k-1)} = \int_a^b L_i^{(k-1)}(x) w(x) dx = \int_a^b \frac{v_k(x)}{(x - x_i) v_k'(x_i)} w(x) dx, \quad i = 1, \dots, k.$$

The computation of the coefficients $A_i^{(k-1)}$ needs the knowledge of the first k quantities

$$c_i = \int_a^b x^i w(x) dx, \quad i = 0, 1, \dots,$$

which are called the **moments** of the linear functional I .

The linear functional I is completely defined as soon as its moments are known.

Whatever be the points x_1, \dots, x_k , the formula is

exact on \mathcal{P}_{k-1} ,

the vector space of polynomials of degree at most $k - 1$,
which means that

$$\forall \mathbf{g} \in \mathcal{P}_{k-1}, \quad \mathbf{I}(\mathbf{R}_{k-1}) = \mathbf{I}(\mathbf{g}).$$

GAUSSIAN QUADRATURES:

Let us now construct a quadrature method with a higher degree of exactness.

A necessary and sufficient condition that the quadrature method becomes **exact on** \mathcal{P}_{2k-1} is that the polynomial v_k , whose zeros are the interpolation points x_1, \dots, x_k , satisfies the conditions

$$\int_a^b x^i v_k(x) w(x) dx = 0, \quad i = 0, \dots, k-1.$$

In that case, v_k is the polynomial of degree k belonging to the family of **orthogonal polynomials** with respect to w on $[a, b]$, it is denoted by P_k , and the preceding conditions are called the **orthogonality conditions**. For all k , such a polynomial P_k exists, and its zeros are real, distinct, and in $[a, b]$, and they interlace with those of P_{k+1} and P_{k-1} .

The quadrature method built on the zeros of P_k as interpolation points is called a

Gaussian quadrature method.

The corresponding coefficients $A_i^{(k-1)}$ are called the

Christoffel numbers.

We now have

$$\forall g \in \mathcal{P}_{2k-1}, \quad \mathbf{I}(\mathbf{R}_{k-1}) = \mathbf{I}(g).$$

ERROR ESTIMATES:

When computing an approximate quantity, an important problem is to estimate its accuracy. Several procedures for estimating the error of Gaussian quadratures exist. Let us describe two of them.

Kronrod procedure

It is possible to estimate the error in a Gaussian quadrature method by using **Kronrod procedure**.

It is a quite general principle in numerical analysis that, when having two different approximations of the same quantity, and when one of them is more accurate than the other, their difference is a good estimation of the error on the less precise approximation.

This is the idea behind Kronrod procedure.

Hence, we will now construct a new quadrature formula, based on $k + n$ interpolation points, and leading to a better approximation of $I(g)$.

The number of evaluations of g has to be minimized. Thus, these $k + n$ points will be chosen as the k zeros of the orthogonal polynomial P_k , where the values of g have already been computed, plus n additional points.

These additional points are chosen in an **optimal way**, that is such that the new quadrature be exact on the vector space of polynomials of the **highest possible degree**.

These n additional points have to be the zeros of the polynomial V_n of degree n satisfying

$$\int_a^b x^i V_n(x) P_k(x) w(x) dx = 0, \quad i = 0, \dots, n - 1.$$

The coefficients of V_n have to be determined from these conditions, which is only possible when $n \geq k$.

For $n = k + 1$, the polynomial V_{k+1} is called a **Stieltjes polynomial** since it was introduced by Thomas Jan Stieltjes in his last letter to Charles Hermite on November 8, 1894, seven weeks before his premature death.

The quadrature formula built on the zeros of P_k and on those of V_{k+1} becomes **exact on \mathcal{P}_{3k+1}** .

But a crucial point has to be mentioned : such a Stieltjes polynomial must have its zeros real, distinct, in $[a, b]$, and they must also be distinct from the zeros of P_k . These conditions are not satisfied for any weight function w and, in this case, Kronrod procedure cannot be applied.

Anti-Gaussian quadratures

Anti-Gaussian quadrature rules were introduced by Laurie. They consist in building a new quadrature rule using $k + 1$ nodes and whose error is precisely the **opposite** to the error of the Gaussian quadrature formula for polynomials of degree at most $2k + 1$.

Denoting by $I_G^{(k)}$ the linear functional corresponding to the Gaussian quadrature on k nodes, and by $I_A^{(k+1)}$ the functional corresponding to this anti-Gaussian formula with $k + 1$ nodes, this idea reads

$$I(p) - I_G^{(k)}(p) = -(I(p) - I_A^{(k+1)}(p)), \quad \forall p \in \mathcal{P}_{2k+1},$$

that is

$$\mathbf{I}_A^{(k+1)}(\mathbf{p}) = \mathbf{2I}(\mathbf{p}) - \mathbf{I}_G^{(k)}(\mathbf{p}), \quad \forall \mathbf{p} \in \mathcal{P}_{2k+1}.$$

The preceding formula means that $I_A^{(k+1)}(g)$ is the Gaussian quadrature formula for the linear functional $I_A^{(k+1)} = 2I - I_G^{(k)}$, and that, in fact, $I(g) \simeq [I_A^{(k+1)}(g) + I_G^{(k)}(g)]/2$, a quadrature formula whose degree of exactness is $2k + 1$. Then, the error of the Gaussian quadrature formula is estimated by

$$I(g) - I_G^{(k)}(g) \simeq [I_A^{(k+1)}(g) - I_G^{(k)}(g)]/2.$$

The anti-Gaussian quadrature formula has to be constructed by computing the orthogonal polynomial of degree $k + 1$ with respect to the linear functional $2I - I_G^{(k)}$, then using its zeros as the nodes of $I_A^{(k+1)}(g)$, and finally computing the corresponding Christoffel numbers.

PADÉ APPROXIMATION OF POWER SERIES

We consider now the formal power series

$$f(t) = c_0 + c_1 t + c_2 t^2 + \dots ,$$

and we define the **linear functional** \mathbf{c} by its moments as

$$c(x^i) = \begin{cases} c_i, & i = 0, 1, \dots \\ 0, & i < 0. \end{cases}$$

Obviously, it holds

$$\mathbf{f}(t) = \mathbf{c} \left(\frac{\mathbf{1}}{\mathbf{1} - \mathbf{x}t} \right),$$

where \mathbf{c} **acts on** \mathbf{x} , and t is considered as a parameter.

Indeed, by the linearity of c , we have

$$\begin{aligned} c\left(\frac{1}{1-xt}\right) &= c(1 + xt + x^2t^2 + \dots) = c(1) + c(x)t + c(x^2)t^2 + \dots \\ &= c_0 + c_1t + c_2t^2 + \dots = f(t). \end{aligned}$$

Thus, the computation of $c(1/(1-xt))$ is similar to the computation of $I(g)$ for the particular function

$$\mathbf{g(x)} = \frac{\mathbf{1}}{\mathbf{1-xt}},$$

where the linear functional I is replaced by c .

Thus, an approximation of $f(t) = c(1/(1-xt))$ can be obtained by replacing $g(x) = 1/(1-xt)$ by its interpolation polynomial R_{k-1} at some points.

PADÉ-TYPE APPROXIMANTS:

Let v_k be a polynomial of exact degree k , and let x_1, \dots, x_n be its distinct zeros, each one with multiplicity k_i , for $i = 1, \dots, n$, and $k_1 + \dots + k_n = k$. Thus

$$v_k(x) = (x - x_1)^{k_1} \cdots (x - x_n)^{k_n}.$$

The polynomial

$$\mathbf{R}_{\mathbf{k}-1}(\mathbf{x}) = \frac{1}{1 - \mathbf{x}\mathbf{t}} \left(1 - \frac{\mathbf{v}_{\mathbf{k}}(\mathbf{x})}{\mathbf{v}_{\mathbf{k}}(\mathbf{t}^{-1})} \right)$$

is the **Hermite interpolation polynomial** of the function $g(x) = 1/(1 - xt)$ at the zeros of v_k . By definition, it satisfies the conditions

$$R_k^{(j)}(x_i) = \frac{d^j}{dx^j} \left(\frac{1}{1 - xt} \right) \Big|_{x=x_i}, \quad i = 1, \dots, n; j = 0, \dots, k_i - 1.$$

Now, as in a quadrature method, $c(R_{k-1}(x))$ will provide an approximation of $c(1/(1 - xt))$.

We obtain

$$\begin{aligned} c(R_{k-1}(x)) &= \frac{1}{t^k v_k(t^{-1})} t^{k-1} c\left(\frac{v_k(t^{-1}) - v_k(x)}{t^{-1} - x}\right) \\ &= \frac{\tilde{w}_k(t)}{\tilde{v}_k(t)}, \end{aligned}$$

where the polynomial w_k is defined by

$$w_k(t) = c\left(\frac{v_k(x) - v_k(t)}{x - t}\right),$$

and $\tilde{v}_k(t) = t^k v_k(t^{-1})$, and, similarly, $\tilde{w}_k(t) = t^{k-1} w_k(t^{-1})$. The polynomials \tilde{v}_k and \tilde{w}_k are obtained from v_k and w_k , respectively, by reversing the numbering of the coefficients.

w_k is a polynomial of degree $k - 1$ in t .

Thus $c(R_{k-1}(x))$ is a rational function in t with a numerator of degree $k - 1$, and a denominator of degree k , and we have

$$\begin{aligned}c(R_{k-1}(x)) &= c\left(\frac{1}{1 - xt}\right) - \frac{1}{v_k(t^{-1})} c\left(\frac{v_k(x)}{1 - xt}\right), \\ &= f(t) - \frac{t^k}{\tilde{v}_k(t)} c\left(\frac{v_k(x)}{1 - xt}\right), \\ &= f(t) + \mathcal{O}(t^k),\end{aligned}$$

which shows that the power series expansion of $c(R_{k-1})$ matches the series f up to the term of degree $k - 1$ inclusively, that is up to the degree of the numerator.

The rational function $c(R_{k-1}(x)) = \tilde{w}_k(t)/\tilde{v}_k(t)$ is called a **Padé-type approximant**

of f , it is denoted by $(k-1/k)_f(t)$, and k is its **order of approximation**, which is equal to the degree of the numerator plus 1. The polynomial v_k , whose choice is free, is called the **generating polynomial** of the approximant.

From this approximant, it is possible to construct Padé-type approximants $(p/q)_f$ with arbitrary degrees in the numerator and in the denominator, and it holds

$$(p/q)_f(t) = f(t) + \mathcal{O}(t^{p+1}).$$

The computation of $(p/q)_f$ needs the knowledge of c_0, \dots, c_p .

The procedure for constructing Padé-type approximants is completely similar to the one leading to interpolatory quadrature formulas.

The same order of exactness is achieved.

The only difference is that the zeros of the generating polynomial v_k do not need to be known, they are no longer restricted to be in an interval of the real line, and they could also be complex and/or multiple.

Thus, Padé-type approximants can be interpreted as **formal interpolatory quadratures** for the linear function c applied to the function $g(x) = 1/(1 - xt)$.

PADÉ APPROXIMANTS:

We want now to **improve the order of approximation** of $(k - 1/k)_f$.

The error formula could be written as

$$f(t) - (k - 1/k)_f(t) = \frac{t^k}{\tilde{v}_k(t)} c(v_k(x)(1 + xt + x^2t^2 + \dots)).$$

If we impose that v_k satisfies the condition $c(v_k(x)) = 0$, then the first term in the expansion of the error disappears, and the order of approximation becomes $k + 1$. If, in addition, we also impose the condition $c(xv_k(x)) = 0$, the second term in the expansion of the error also disappears, and the order of approximation becomes $k + 2$, and so on.

Let us look for the **maximal order of approximation** which could be achieved.

The polynomial v_k depends of $k + 1$ coefficients but, on the other side, a rational function is defined apart a multiplying factor in its numerator and its denominator. Thus, v_k depends only on k coefficients and, so, we can only add the k additional conditions

$$\mathbf{c}(\mathbf{x}^i \mathbf{v}_k(\mathbf{x})) = \mathbf{0}, \quad \mathbf{i} = \mathbf{0}, \dots, \mathbf{k} - \mathbf{1}.$$

These conditions are called the **orthogonality conditions**, and the polynomial v_k , which will now be denoted by P_k , is the k th member of the family of **formal orthogonal polynomials** (FOP) with respect to the linear functional c . The associated polynomial of $v_k \equiv P_k$ will now be denoted by Q_k instead of w_k .

We immediately see that

$$c(R_{k-1}(x)) = \frac{\tilde{Q}_k(t)}{\tilde{P}_k(t)} = f(t) - \frac{t^{2k}}{\tilde{P}_k(t)} c\left(\frac{x^k P_k(x)}{1 - xt}\right) = f(t) + \mathcal{O}(t^{2k}).$$

The error is given by

$$f(t) - [k - 1/k]_f(t) = \frac{t^{2k}}{\tilde{P}_k(t)} c\left(\frac{x^k P_k(t)}{1 - xt}\right).$$

Now, our Padé-type approximant achieves the **order of approximation $2k$** (= the degree of the numerator + the degree of the denominator + 1), **instead of k** , it is called a

Padé approximant

of f , and it will be denoted by $[k - 1/k]_f(t)$ (notice the square brackets instead of the round parenthesis).

Similarly to what was done for Padé-type approximants, it is possible to construct Padé approximants with arbitrary degrees in the numerator and in the denominator, and such that $[p/q]_f(t) = f(t) + \mathcal{O}(t^{p+q+1})$. The computation of $[p/q]_f$ needs the knowledge of c_0, \dots, c_{p+q} .

The procedure for constructing Padé approximants is the same as the process leading to a Gaussian quadrature formula. Again, the order of exactness is the same. The zeros of the FOPs are not subject to any restriction, as in the Gaussian quadrature methods.

Thus, Padé approximants can be interpreted as **formal Gaussian quadrature methods** for the linear function c applied to the function $g(x) = 1/(1 - xt)$.

Conditions have to be imposed on c for the orthogonal polynomials P_k to exist, which was not the case before.

ERROR ESTIMATES:

Since Padé approximants could be understood as formal Gaussian quadrature methods, their error could be estimated by Kronrod procedure or by constructing anti-Gaussian Padé approximants.

Kronrod procedure

For estimating the error, we will add, as we did before, n new interpolation points to the k zeros of P_k .

In other terms, we will construct the Padé-type approximant $(n+k-1/n+k)_f$ with the generating polynomial $v(x) = P_k(x)V_n(x)$, where V_n is a polynomial of degree n whose zeros are the n new interpolation points.

Again, the smallest possible value for n is $n = k + 1$.

In order to have a Padé-type approximant with the highest possible order of approximation, we will, as in Kronrod procedure, choose V_{k+1} such that

$$\mathbf{c}(\mathbf{x}^i \mathbf{P}_k(\mathbf{x}) \mathbf{V}_{k+1}(\mathbf{x})) = \mathbf{0}, \quad \mathbf{i} = \mathbf{0}, \dots, \mathbf{k}.$$

Thus, we obtain an approximant $(2k/2k+1)_f$ satisfying $f(t) - (2k/2k+1)_f(t) = \mathcal{O}(t^{3k+2})$, which is exactly the result of Kronrod quadrature formula to be exact on the vector space of polynomials of degree at most $3k+1$.

The difference $(2k/2k+1)_f(t) - [k-1/k]_f(t)$ is a good estimation of the error $f(t) - [k-1/k]_f(t)$. Indeed, we have

$$\frac{\mathbf{f}(\mathbf{t}) - [\mathbf{k} - \mathbf{1}/\mathbf{k}]_{\mathbf{f}}(\mathbf{t})}{(2\mathbf{k}/2\mathbf{k} + \mathbf{1})_{\mathbf{f}}(\mathbf{t}) - [\mathbf{k} - \mathbf{1}/\mathbf{k}]_{\mathbf{f}}(\mathbf{t})} = \mathbf{1} + \mathbf{t}^{\mathbf{k}+2} \frac{\mathbf{1}}{\mathbf{c}(\mathbf{x}^{\mathbf{k}} \mathbf{P}_{\mathbf{k}})} \mathbf{c} \left(\frac{\mathbf{x}^{\mathbf{k}+1} \mathbf{P}_{\mathbf{k}}(\mathbf{x}) \mathbf{V}_{\mathbf{k}+1}(\mathbf{x})}{\mathbf{1} - \mathbf{x}\mathbf{t}} \right).$$

From the practical point of view, it is not necessary to know the zeros of V_{k+1} (nor those of P_k), contrary to the case of definite integrals, and there is no limitation to the use of the procedure if the zeros are complex and/or multiple, and they no longer have to belong to the interval of integration (since no interval arises in Padé-type approximation).

Anti-Gaussian Padé approximants

There exists a linear functional d , which depends on v_k (or P_k), such that

$$d\left(\frac{1}{1-xt}\right) = c(R_{k-1}(x)) = (k - 1/k)_f(t) \quad (\text{or } [k - 1/k]_f(t)).$$

In the case of Padé-type approximation (arbitrary v_k), this linear functional d plays the role of the functional leading to an interpolatory quadrature formula with arbitrary nodes or, for Padé approximation ($v_k \equiv P_k$), of the functional $I_G^{(k)}$ used above.

As in the case of definite integrals, anti-Gaussian Padé-type or Padé approximants, denoted by $((k/k + 1))_f$, can be defined as

$$((k/k + 1))_f(t) = (2c - d) \left(\frac{1}{1 - xt} \right).$$

The moments of the linear functional d can be recursively computed.

The error of the Padé-type or the Padé approximant is given, respectively, by

$$(c - d) \left(\frac{1}{1 - xt} \right) = \mathcal{O}(t^{k+1}) \text{ or } \mathcal{O}(t^{2k+2}).$$

THE ε -ALGORITHM:

When a sequence of numbers (S_n) is slowly converging (or even diverging), it is sometimes possible to modify the way it is built. In some other cases, one has no access to the formation of its members. Then, one can try to transform the sequence into another one converging faster to the same limit under some assumptions.

A quite powerful and well-known sequence transformation is due to Shanks. It consists in building the set of sequences

$$e_k(S_n) = H_{k+1}(S_n)/H_k(\Delta^2 S_n), \quad k, n = 0, 1, \dots$$

where $\Delta^2 S_n = S_{n+2} - 2S_{n+1} + S_n$, and $H_k(u_n)$ denotes the Hankel determinant. When $k = 1$, Aitken's Δ^2 process is recovered.

The numbers $e_k(S_n)$ can be recursively computed by the ε -algorithm of Wynn

$$\begin{aligned}\varepsilon_{-1}^{(n)} &= 0, & \varepsilon_0^{(n)} &= S_n, & n &= 0, 1, \dots \\ \varepsilon_{k+1}^{(n)} &= \varepsilon_{k-1}^{(n+1)} + (\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)})^{-1}, & k, n &= 0, 1, \dots\end{aligned}$$

It holds $\varepsilon_{2k}^{(n)} = e_k(S_n)$.

When the ε -algorithm is applied to the partial sums of a formal power series f , that is when

$$S_n = \sum_{i=0}^n c_i t^i, \quad n = 0, 1, \dots$$

then

$$\varepsilon_{2k}^{(n)} = [n + k/k]_f(t).$$

Thus, this algorithm can be used for computing the Padé approximants recursively.

PERSPECTIVES

For the approximate computation of definite integrals, **Clenshaw–Curtis quadrature formulas** are often almost as precise as Gaussian quadrature methods. They make use of the extrema (including the endpoint ones) of the Chebyshev polynomial T_{k-1} , that is $x_i = \cos((i-1)\pi/(k-1))$, $i = 1, \dots, k$. An interesting problem will be to study the Padé-type approximants based on the same points.

It could also be interesting to study Padé-type approximants based on other sets of points, as those used in Gauss–Radau, Gauss–Lobatto, or Féjer formulas.

The **conjugate gradient algorithm** for solving a system of linear equations with a symmetric positive definite matrix can be interpreted in terms of a Gaussian quadrature formula for a certain positive functional. Therefore, its error could be estimated by the methods described above.

Gauss–Radau and Gauss–Lobatto quadrature formulas have also been used for this purpose. For arbitrary matrices, Lanczos method (which can be implemented, for example, by the bi-conjugate gradient algorithm) can be interpreted via Padé approximants, that is via formal Gaussian quadrature methods, and it is related to the topological ε -algorithm.

General extrapolation procedures for estimating the error in the solution of systems of linear equations have been proposed. Anti-Gaussian quadratures rules are also useful in this context. Another interesting problem will be to study a possible use of Kronrod procedure for the same purpose.

Several of the ideas described in this paper have been extended to the case of **vector Padé approximation**, and to **series of functions**.