CONVERGENCE ACCELERATION OF ALTERNATING SERIES

Henri Cohen, Fernando Rodriguez Villegas and Don Zagier

Abstract. We discuss some linear acceleration methods for alternating series which are in theory and in practice much better than that of Euler-Van Wijngaarden. One of the algorithms, for instance, permits one to calculate $\sum(-1)^k a_k$ with an error of about 17.93^{-n} from the first n terms for a wide class of sequences $\{a_k\}$. Such methods are useful for high precision calculations frequently appearing in number theory.

The goal of this paper is to describe some linear methods to accelerate the convergence of many alternating sums. The main strength of these methods is that they are very simple to implement and permit rapid evaluation of the sums to the very high precision (e.g. several hundred digits) frequently occurring in number theory.

The typical series we will be considering are alternating series $S = \sum_{k=0}^{\infty}(-1)^k a_k$, where a_k is a reasonably well-behaved function of k which goes slowly to 0 as $k \to \infty$. Assume we want to compute a good approximation to S using the first n values a_k. Then our first algorithm is

Algorithm 1.

Initialize: $d = (3+\sqrt{8})^n$; $d = (d + 1/d)/2$; $b = -1$; $c = -d$; $s = 0$;

For $k = 0$ up to $k = n-1$, repeat the following:

$c = b - c$; $s = s + c \cdot a_k$; $b = (k+n)(k-n)b/((k+1/2)(k+1))$;

Output: s/d.

This algorithm computes an approximation to S as a weighted sum of $a_0, \dots, a_{n-1}$ with universal rational coefficients $c_{n,k}/d_n$ ($= c/d$ in the notation of the algorithm; note that both c and d are integers). For instance, for $n = 1$, 2, 3, 4 the approximations given by the algorithm are $2a_0/3$, $(16a_0 - 8a_1)/17$, $(98a_0 - 80a_1 + 32a_2)/99$, and $(576a_0 - 544a_1 + 384a_2 - 128a_3)/577$, respectively. The denominator d_n grows like 5.828^n and the absolute values of the coefficients $c_{n,k}$ decrease smoothly from $d_n - 1$ to 0. Proposition 1 below proves that for a large class of sequences $\{a_k\}$ the algorithm gives an approximation with a relative accuracy of about 5.828^{-n}, so that to get D decimal digits it suffices to take n equal to approximately $1.31D$. Notice that the number of terms and the time needed to get a given accuracy are essentially independent of the particular series being summed, if we assume that the a_k's themselves are either easy to compute or have been precomputed; on a Sparcstation 5 using Pari, for instance, the computation of S to 100 or 1000 decimal

Key words and phrases. Convergence acceleration, alternating sum, Chebyshev polynomial.
1991 AMS Classification 11Y40

digits requires about .1 or 6 seconds, respectively. The algorithm uses $O(1)$ storage and has a running time of $O(1)$ per value of a_k used.

Proposition 1. *For integers $n \geq k \geq 0$ set*

$$d_n = \frac{(3+\sqrt{8})^n + (3-\sqrt{8})^n}{2}$$

and

$$c_{n,k} = (-1)^k \left(d_n - \sum_{m=0}^{k} \frac{n}{n+m} \binom{n+m}{2m} 2^{2m} \right) = (-1)^k \sum_{m=k+1}^{n} \frac{n}{n+m} \binom{n+m}{2m} 2^{2m} .$$

Assume that the a_k are the moments of a positive measure on $[0,1]$ and set

$$S = \sum_{k=0}^{\infty} (-1)^k a_k, \qquad S_n = \sum_{k=0}^{n-1} \frac{c_{n,k}}{d_n} a_k .$$

Then

$$|S - S_n| \leq \frac{S}{d_n} \approx \frac{2\,S}{(3+\sqrt{8})^n} .$$

Proof. By assumption, we have

$$a_k = \int_0^1 x^k \, d\mu .$$

where $d\mu$ is a positive measure. Then

$$S = \sum_{k=0}^{\infty} (-1)^k a_k = \int_0^1 \frac{1}{1+x} \, d\mu ,$$

the interchange of summations being justified by the positivity. Let $\{P_n(x)\}$ be a sequence of polynomials such that P_n has degree n and $d_n := P_n(-1) \neq 0$. Set

$$\frac{P_n(-1) - P_n(x)}{1+x} = \sum_{k=0}^{n-1} c_{n,k} \, x^k .$$

Define S_n as in the proposition but with these values of d_n and $c_{n,k}$. Then

$$S_n = \frac{1}{P_n(-1)} \int_0^1 \frac{P_n(-1) - P_n(x)}{1+x} \, d\mu = S - R_n ,$$

say, and by virtue of the positivity of $d\mu$ we can estimate the "error term" R_n by

$$|R_n| \leq \frac{M_n}{|P_n(-1)|} \int_0^1 \frac{1}{1+x} \, d\mu = \frac{M_n}{|P_n(-1)|} \, S ,$$

where M_n is the supremum of $|P_n(x)|$ on $[0, 1]$. It follows that $M_n/|P_n(-1)|$ is an upper bound for the relative error made by approximating S by S_n.

We now choose for $P_n(X)$ the polynomials defined by

$$P_n(\sin^2 t) = \cos 2nt\,,$$

so that $P_n(x) = T_n(1 - 2x)$ where $T_n(x)$ is the ordinary Chebyshev polynomial. Clearly $M_n = 1$ and $P_n(-1) = d_n$. The recursion $P_{n+1}(X) = 2(1-2X)P_n(X) - P_{n-1}(X)$ implies by induction the explicit formula

$$P_n(X) = \sum_{m=0}^{n} (-1)^m \frac{n}{n+m} \binom{n+m}{2m} 2^{2m} X^m\,, \tag{1}$$

so the coefficients $c_{n,k}$ are given as in Proposition 1. This completes the proof of the proposition. Algorithm 1 is simply an efficient way to compute the coefficients $c_{n,k}$ successively.

Remarks.

(1) The method implicit in Proposition 1 is well known in numerical analysis under the heading of "Padé type approximation" (cf. [B], [E], [G], [W]). As we mentioned above, we are concerned with calculations to a high degree of accuracy, where the number of digits gained per number of steps, and the amount of storage required, are crucial. Thus our emphasis is different from that in numerical analysis, where one usually works in fixed, and relatively low, precision. The implementation of Algorithm 1 is good in both respects.

(2) The classical algorithms which are found in the literature (see [PFTV]) are Euler's method or Euler-Van Wijngaarden's method. These can be shown to correspond respectively to the polynomials $P_n(X) = (1 - X)^n$ with convergence like 2^{-n} and polynomials $P_n(X) = X^a(1 - X)^b$ with $a + b = n$ dependent on the particular sequence, with convergence like 3^{-n} for $a = n/3$. Note that a direct implementation of the algorithm given in [PFTV] needs a lot of auxiliary storage if we want high accuracy, while our method does not.

(3) Algorithm 1 computes "on the fly" the coefficients of the polynomial $(P_n(-1) - P_n(X))/(1+X)$, where $P_n(X) = T_n(1-2X)$. Equivalently, we could also compute on the fly only the coefficients of the polynomial $P_n(X)$ itself and use the partial sums of the alternating series instead of the individual terms, using

$$\sum_{k=0}^{n} c_{n,k} a_k = \sum_{m=1}^{n} \frac{n}{n+m} \binom{n+m}{2m} 2^{2m} \sum_{k=0}^{m-1} (-1)^k a_k\,.$$

This can be particularly useful when the sequence of partial sums is naturally given, and not the a_k themselves, as in the continued fraction example mentioned at the end of this paper.

(4) The hypothesis that the a_n's are moments of a positive measure on the interval [0,1] is a well known one, and is equivalent by a famous theorem of Hausdorff

[H] to the *total monotonicity* of the sequence $\{a_n\}$, in the sense that for each fixed k, the sequence $\{\Delta^k a_n\}$ of the k-th forward differences of $\{a_n\}$ has constant sign $(-1)^k$.

(5) In addition to this, trivial modifications are possible. For example, one can replace the step $d = (d + 1/d)/2$ by the simpler one $d = d/2$, since this modifies the final result by a negligible amount, but leaves d as a nonintegral value. We could also immediately divide by d (initializing b to $-1/d$ and c to -1). In our implementation each of these modifications led to slower programs.

(6) We proved convergence of the algorithm (at a rather fast geometric rate) under the above condition. However, examples show that it can be applied to a much wider class of series, and also, as is usual in acceleration methods, to many divergent series.

(7) The choice of the Chebyshev polynomial can be shown to be close to optimal if we estimate the remainder term R_n crudely as we did above. On the other hand, as we will see below, for a different class of alternating series, we can estimate R_n more precisely and find much better polynomials P_n. The corresponding algorithms and their analysis seem to be new.

(8) If the sequence a_k already converges at a geometric rate, better algorithms are possible which are trivial modifications of the ones presented in this paper. For example, if one knows in advance that $-\ln(a_k) \sim k\ln(z)$ for some $z \geq 1$, then in Algorithm 1 simply replace $3 + \sqrt{8}$ by $2z + 1 + 2\sqrt{z(z+1)}$ and multiply by z the right hand side of the recursion formula for b. The convergence is in $(2z + 1 + 2\sqrt{z(z+1)})^{-n}$, and thus faster than the direct application of Algorithm 1. Similar modifications are valid for the other algorithms of this paper. The details are left to the reader.

Examples. To illustrate Algorithm 1, we give a few examples. These examples can also be treated individually using specific methods.

1) By taking logarithms we can compute the product

$$A := \prod_{n=1}^{\infty} \frac{\Gamma(1 + \frac{1}{2n-1})}{\Gamma(1 + \frac{1}{2n})} = 1.0621509055\ldots$$

rapidly to high accuracy. Note that the product is slowly convergent and that the gamma function is hard to compute, so our method is very useful here. In fact as we will see below, Algorithm 2_B is even better in this case.

2) Similar remarks apply to the sum

$$B := \sum_{n=2}^{\infty} (-1)^n \operatorname{Li}_2\left(\frac{2}{n}\right) = 1.1443442096\ldots$$

where $\operatorname{Li}_2(x) = \sum_{k=1}^{\infty} x^k/k^2$ is the dilogarithm function. This sum arose in connection with the computation of a certain definite integral related to Khinchin's constant [BC].

3) The Riemann zeta function can be calculated for reasonable values of the argument by

$$\left(1-2^{1-s}\right)\zeta(s)=\sum_{n=0}^{\infty}\frac{(-1)^n}{(n+1)^s}\,,$$

and we find values like $\zeta(1/2)=-1.4603545088\ldots$ or even

$$\zeta(-1+i)=0.0168761517\ldots-0.1141564804\ldots\,i$$

to high accuracy and very quickly (comparable to Euler-Maclaurin). Note that the latter example works even though the coefficients a_k of our "alternating" series $\sum(-1)^k a_k$ are not alternating or even real and do not tend to zero. The derivatives of $\zeta(s)$ can also be calculated similarly, e.g. we can compute

$$C_1:=\sum_{n=1}^{\infty}\frac{(-1)^n\log n}{\sqrt{n}}=0.1932888316\ldots$$

by Algorithm 1, and then use the identity $C_1=(1-\sqrt{2})\,\zeta'(\frac{1}{2})+\sqrt{2}\,\ln 2\,\zeta(\frac{1}{2})$ to deduce the value of $\zeta'(1/2)=-3.9226461392\ldots$. In a similar manner, we can compute

$$C_2=\sum_{n=1}^{\infty}\frac{(-1)^n\log n}{n}=0.1598689037\ldots$$

by Algorithm 1, and then use the identity $C_2=\log 2(\gamma-1/2\log 2)$ to deduce the value of Euler's constant $\gamma=0.5772156649\ldots$.

Moreover, as suggested to us by R. Sczech, we may even use Algorithm 1 to sum the divergent series

$$\sum_{n=1}^{\infty}(-1)^n\log n\,,$$

recovering the known value $-\frac{1}{2}\log(\frac{\pi}{2})=-0.2257913526\ldots$ of the derivative of $(1-2^{1-s})\zeta(s)$ at $s=0$.

Note that in the first two examples, a_k was in fact the restriction at points of the form $1/k$ of a function analytic in the neighborhood of 0, and so other techniques could be used such as expanding explicitly a_k in powers of $1/k$. However, in the last examples, this cannot be applied.

If we make different assumptions on the sequence $\{a_k\}$, we can use better polynomials than the Chebyshev polynomials. For example, we can reason as follows. From the proof of Proposition 1, the remainder R_n is given by

$$R_n=\int_0^1\frac{P_n(x)}{P_n(-1)(1+x)}\,d\mu\,.$$

Assume that $d\mu = w(x)dx$ for a smooth function $w(x)$. Then taking $P_n(X)$ and d_n as in Proposition 1 and setting $x = \sin^2 t$, we get

$$d_n R_n = \int_0^{\pi/2} \cos(2nt) \frac{w(\sin^2 t)\, \sin 2t}{1 + \sin^2 t}\, dt = \int_0^{\pi/2} \cos(2nt)\, h(t)\, dt$$

for a smooth function $h(t)$. If we integrate twice by parts, we get

$$d_n R_n = \frac{1}{4n^2}\big((-1)^n c_1 + c_2\big) - \frac{1}{4n^2} \int_0^{\pi/2} \cos(2nt)\, h''(t)\, dt\,,$$

for some constants $c_1 = h'(\pi/2)$ and $c_2 = -h'(0)$. If we multiply both sides of this equation by n^2, then the first term $\frac{1}{4}((-1)^n c_1 + c_2)$ has period 2. This suggests replacing the polynomial $P_n(X)$ by the polynomial

$$P_n^{(1)}(X) = n^2 P_n(X) - (n-2)^2 P_{n-2}(X)\,.$$

It is then easily seen that for the new remainder $R_n^{(1)}$ we have $d_n R_n^{(1)} = \mathrm{O}(1/n^5)$ instead of $\mathrm{O}(1/n^2)$. Hence for our purposes $P_n^{(1)}$ is a better polynomial than P_n. Notice also that

$$P_n^{(1)}(\sin^2 t) = \frac{1}{2} \frac{d^2}{dt^2} \big(\sin(2t) \sin(2(n-1)t)\big)\,.$$

Continuing in this way, we define a double family of polynomials $P_n^{(m)}$ as the m-th difference (with step 2) of the sequence $n^{m+1} P_n$ (where we set $P_{-n} = P_n$ for $n > 0$ since $\cos(2nt)$ is an even function of n), and find that for $m > 0$

$$P_n^{(m)}(\sin^2 t) = \frac{1}{2} \Big(\frac{d}{dt}\Big)^{m+1} \big(\sin(2t)^m \sin(2(n-m)t)\big)$$

The corresponding remainder term $R_n^{(m)}$ satisfies $d_n R_n^{(m)} = \mathrm{O}(1/n^{2m+2})$ (and even $\mathrm{O}(1/n^{2m+3})$ if m is odd) as $n \to \infty$ for fixed m, so we get better and better sequences of polynomials.

As polynomials in X, these polynomials can be computed either from the formula giving the m-th difference, i.e.

$$P_n^{(m)}(X) = \sum_{0 \le r \le m} (-1)^r \binom{m}{r} (n - 2r)^{m+1} P_{n-2r}(X)$$

(where we naturally set $P_{-n}(X) = P_n(X)$) or by the formulas

$$P_n^{(m)}(X) = 2^{m+r-1} (\delta D)^r\, \delta^s \left(\sum_{k=0}^{n-m-1} (-1)^k \binom{2n-2m}{2k+1} X^{k+r} (1-X)^{n-m-1-k+r} \right)$$

where $m+1=2r+s$ with $0 \le s \le 1$, $D = \dfrac{d}{dX}$, $\delta = 2X(1-X)\dfrac{d}{dX} + 1 - 2X$.

This suggests taking either the diagonal sequence $A_n = P_n^{(n-1)}/(2^{n-1}n!)$ or the sequence $B_n = P_n^m/((n-m)(m+1)!2^m$ with $m = \lfloor n/2 \rfloor$, where the (unimportant) normalizing constants have been chosen so that $A_n(0) = B_n(0) = 1$ (other ratios between m and n could be considered and we will briefly comment on this later).

The first few values are $A_0(X) = 1$, $A_1(X) = 1 - 2X$, $A_2(X) = 1 - 8X + 8X^2$, $A_3(X) = 1 - 20X + 54X^2 - 36X^3$ and $B_n(X) = A_n(X)$ for $n \le 4$.

Note that the formulas giving the polynomials A_n are particularly simple: we have

$$A_n(\sin^2 t) = \frac{1}{2^n n!}\frac{d^n}{dt^n}\left(\sin^n 2t\right),$$

and

$$A_n(X) = \frac{2^r}{(2r)!}\left((1-2X)\frac{d}{dX} + 2(X-X^2)\frac{d^2}{dX^2}\right)^r\left(X^r(1-X)^r(1-2X)^s\right)$$

if $n = 2r+s$ with $0 \le s \le 1$.

Algorithm 1 can be immediately generalized to any sequence Q_n of polynomials, except that the coefficients of Q_n cannot be computed on the fly, thus giving rise to the following family of algorithms.

Algorithm 2_Q. Let $Q_n(X) = \sum_{k=0}^{n} b_k X^k$.
Set $d = Q_n(-1)$; $c = -d$;
For $k = 0$ up to $k = n-1$, repeat the following:
  $c = b_k - c$; $s = s + c \cdot a_k$;
Output: s/d.

In particular, applying this to the families $Q_n = A_n$ and $Q_n = B_n$, we obtain two algorithms 2_A and 2_B which are of the same sort as Algorithm 1 in that they output an approximation S_n which is a universal rational linear combination of $a_0, \ldots, a_{n-1}$. The values for $n=1$ and $n=2$ are $2a_0/3$, $(16a_0 - 8a_1)/17$ as before, while those for $n=3$ and $n=4$ are $(110a_0 - 90a_1 + 36a_2)/111$ and $(2288a_0 - 2168a_1 + 1536a_2 - 512a_3)/2191$, respectively (since $A_n = B_n$ for $n \le 4$, the coefficients are the same for both algorithms up to that point).

Let us analyze the speed of convergence of these algorithms. For Algorithm 2_A we will find that it is like 7.89^{-n} for a large class of sequences $\{a_k\}$ and like 17.93^{-n} for a smaller class, both better than the 5.83^{-n} we had for Algorithm 1.

For the same two classes of sequences, Algorithm 2_B will be like 9.56^{-n} and 14.41^{-n}. In other words, depending on the sequence Algorithm 2_A or 2_B may be the better choice.

On the other hand, unlike Algorithm 1, we do not have a quick way to compute the individual coefficients $c_{n,k}$ in time $O(1)$ each but must compute (and store) the whole polynomial $A_n(X)$ or $B_n(X)$. As a result, these algorithms require storage $O(n)$ and time $O(n^2)$ instead of $O(1)$ and $O(n)$ as before. Thus they are inferior to Algorithm 1 if the numbers a_k are easy to compute (like $a_k = 1/(k+1)$), but superior to it if the main part of the running time is devoted to the computation of the a_k (as in Examples 1 or 2 above), or in the extreme case when we only know a fixed number of values $a_0, a_1, \ldots, a_{n-1}$ and want to make the best guess of the value of $\sum(-1)^k a_k$ on the basis of this data.

To state the result we need the constants defined by

$$\beta_A = \cosh\alpha_A = 7.8898395\ldots, \qquad \beta_B = e^{\alpha_B} = 9.55699\ldots,$$
$$\gamma_A = \beta_A r_0 = 17.9340864\ldots, \qquad \gamma_B = \beta_B\sqrt{r_0} = 14.408763\ldots.$$

Here $\alpha_A = 2.754682\ldots$ and $\alpha_B = 2.2572729\ldots$ are the unique real roots of $\alpha_A - \tanh\alpha_A = 2L$ and $\alpha_B - e^{-\alpha_B}\sinh\alpha_B = 2L$, respectively, where $L = \log(1+\sqrt{2})$, and $r_0 = \sqrt{4t_0/\sin 4t_0} = 2.27306\ldots$ where $t_0 = 0.652916\ldots$ is the unique root in $[0, \pi/2]$ of $t_0^2 + L^2 = \frac{1}{2}t_0\,\tan 2t_0$.

In addition, let CA be the union of the images in the complex plane of the two maps

$$t \mapsto \sin^2\left(t - (\sin(2t)/2)(\cos(2t) \pm i\sin(2t))\right) \quad \text{for } 0 \le t \le \pi/2\ ,$$

and let CB be the union of the images in the complex plane of the two maps

$$t \mapsto \sin^2\left(t - (r_0\sin(2t)/2)\left(r_0\cos(2t) \pm i\sqrt{1 - r_0^2\cos(2t)^2}\right)\right) \quad \text{for } t_0 \le t \le \pi/2 - t_0$$

(see Figure 1). Then we have the following result.

Proposition 2. *Assume that the coefficients a_k are given as moments $a_k = \int_0^1 w(x)\,x^k\,dx$. For $Q = A$ or $Q = B$, let $R_n^Q = S - S_n^Q$ be the difference between the true value of S and the output S_n^Q of Algorithm 2_Q.*

(1) *If $w(x)$ extends analytically to the interior of the region bounded by the Curve CA, then for $Q = A$ or $Q = B$, $|R_n^Q|$ is bounded by $(\beta_Q + o(1))^{-n}$.*
(2) *If $w(x)$ extends analytically to the whole complex plane (or only to the interior of the region bounded by the Curve CB), then for $Q = A$ and $Q = B$, $|R_n^Q|$ is bounded by $(\gamma_Q + o(1))^{-n}$*

In fact the proof will show that to get convergence like γ_Q^{-n} we can allow singularities $x^{m-1/2}$ ($m \ge 0$) at the origin (i.e., it is sufficient if $xw(x^2)$ rather than $w(x)$ itself be analytic). If w is analytic in a smaller region than those described in Proposition 2, then we still get exponential convergence of S_n^Q to S, but with a worse constant.

Here are a few simple examples. We set $\beta = \beta_Q$ and $\gamma = \gamma_Q$ with $Q = A$ or $Q = B$, depending on whether Algorithm 2_A or 2_B is used.

1. $a_k = 1/(k+1)$, $S = \log 2$. Here $w(x) = 1$, so the proposition applies directly to give speed of convergence γ^{-n}. Hence in this case Algorithm 2_A is better than Algorithm 2_B,

2. $a_k = 1/(2k+1)$, $S = \pi/4$. Here $w(x) = \frac{1}{2}x^{-1/2}$, so $xw(x^2)$ is analytic and we again get convergence γ^{-n}. Once again Algorithm 2_A is better.

3. $a_k = 1/(3k+1)$, $S = (\log 2 + \pi/\sqrt{3})/3$. Here $w(x) = \frac{1}{3}x^{-2/3}$ with a singularity at 0, so the convergence is like β^{-n}. Hence in this case Algorithm 2_B is better than Algorithm 2_A.

4. $a_k = 1/(k+1)^2$, $S = \pi^2/12$. Here $w(x) = \log(1/x)$, again with a singularity at 0, so we get β^{-n} convergence. The same applies to $a_k = 1/(k+1)^s$, where $w(x)$ is proportional to $\log^{s-1}(1/x)$ and the convergence is again like β^{-n}. Again Algorithm 2_B is better.

5. $a_k = 1/(k^2+1)$, $S = 1/2 + \pi/(e^\pi - e^{-\pi})$. Here $w(x) = \sin(\log(1/x))/x$, again with convergence like β^{-n}. Again Algorithm 2_B is better.

Proof. We first consider the case of Algorithm 2_A. The first thing we need to know is the asymptotic behavior of $A_n(-1)$. By Lagrange's formula we have

$$\sum_{n=0}^{\infty} A_n(\sin^2 t)\, u^n = \frac{1}{1 - u\cos 2z}\bigg|_{z - \frac{u}{2}\sin 2z = t} \tag{2}$$

and in particular

$$\sum_{n=0}^{\infty} A_n(-1)\, u^n = \frac{1}{1 - u\cos 2z}\bigg|_{z - \frac{u}{2}\sin 2z = iL},$$

where $L = \log(\sqrt{2}+1)$ (or more precisely, any value $\pm\log(\sqrt{2}+1) + in\pi$). To find the limiting value of $|A_n(-1)|^{1/n}$, we need to look at the values of u for which the expression on the right becomes singular. This clearly happens if $u\cos 2z = 1$ for a value of z with $z - (u/2)\sin 2z = iL$. The smallest value of $|u|$ occurs for $z = i\alpha_A/2$ and equals $1/\beta_A$, with α_A and β_A as in the proposition. Hence $\lim |A_n(-1)|^{1/n} = \beta_A$. A more careful analysis gives $A_n(-1) \sim 2\beta_A^{n+1/2}/\sqrt{\pi n(\beta_A^2 - 1)}$ or even an asymptotic expansion, but we will not use this.

Now if we used the proof Proposition 1 directly then the error term R_n^A would be estimated essentially by $M_n/A_n(-1)$, where $M_n = \max\limits_{0\le x\le 1} |A_n(x)|$. Using (2), one can show that this number grows like $(1.5088\ldots)^n$ (the maximum is attained at $x = 1/2$ if n is even), leading to an error estimate of about 5.23^{-n}, which is worse than we had before. But of course the motivation for introducting the polynomials $P_n^{(m)}$ and the diagonal subsequence A_n was to improve the error term by assuming that the function $w(x)$ or $h(t)$ was smooth and use repeated integration by parts. Making this assumption and doing the integration by parts, we obtain

$$\begin{aligned} A_n(-1)\, R_n^A &= \int_0^1 A_n(x)\,\frac{w(x)}{1+x}\,dx = \int_0^{\pi/2} A_n(\sin^2 t)\, h(t)\, dt \\ &= \frac{1}{2^n n!}\int_0^{\pi/2} \frac{d^n}{dt^n}\big(\sin^n(2t)\big)\, h(t)\, dt \\ &= \frac{(-1)^n}{2^n n!}\int_0^{\pi/2} \sin^n(2t)\,\frac{d^n h(t)}{dt^n}\, dt\,, \end{aligned}$$

so by Taylor's theorem

$$\sum_{n=0}^{\infty} A_n(-1)\, R_n^A\, u^n = \int_0^{\pi/2} h\big(t - \frac{u}{2}\sin 2t\big)\, dt\,.$$

As t goes from 0 to $\pi/2$ for a fixed (complex) value of u, the argument $t - \frac{u}{2}\sin 2t$ moves along a path C_u connecting the points 0 and $\pi/2$. If for some $r > 0$ the function $h(t)$ is entire in the region $D(r) = \bigcup_{|u|\le r} C_u$, then the above integral representation of $\sum A_n(-1)\, R_n^A\, u^n$ shows that this sum is analytic in the disc $|u| < r$ and hence that $\lim_n |A_n(-1)R_n^A|^{1/n}$ is at most $1/r$. The best value of r we can hope for (unless w is very special) is the number $r_0 = 2.27306\ldots$ for which the point $t = iL$ lies on the boundary of $D(r)$, since at this point the denominator $1+\sin^2 t$ of $h(t)$ will vanish. (The value of r_0 can be computed by simple calculus: r_0 is the minimum of $2\sqrt{t^2+L^2}/|\sin 2t|$ for $t \in [0, \pi/2]$ and is equal to $\sqrt{4t_0/\sin 4t_0}$ with t_0 as in the proposition.) This then leads to the estimate $\lim |R_n^A|^{1/n} \le \gamma_A^{-1}$ with $\gamma_A = r_0\beta_A$, and is valid whenever h is analytic in $D(r_0)$ or equivalently, whenever w is analytic in the region $\{\sin^2 t \mid t \in D(r_0)\}$, which is the region bounded by Curve CB in Figure 1. If $w(x)$ has a singularity within this region then we get a worse estimate. In particular, if $w(x)$ has a singularity at the origin, as in the examples 3–6 above and many others, then the convergence of S_n^A to S will be like β_A^{-n} if $h(t)$ has no singularities in $D(1)$, since $r = 1$ is the largest value of r for which 0 is not in the interior of $D(r)$. The boundary of the region $\{\sin^2 t \mid t \in D(1)\}$ is shown as Curve CA in Figure 1.

The case of Algorithm 2_B is very similar, but slightly more complicated. We sketch it briefly. To be able to apply Lagrange's formula, we introduce

$$C_m(t) = \frac{1}{m!2^m}\left(\frac{d}{dt}\right)^m \left((e^{2it}\sin 2t)^m \frac{e^{-2it}}{\sin 2t}\right)$$

so that

$$B_{2m}(\sin^2 t) = \frac{C_{m+1}(t) + C_{m+1}(-t)}{2im} \quad \text{for } m > 0\,.$$

Then a similar use of Lagrange's formula shows that $\lim |C_m(\pm iL)|^{1/m} = \beta_B^2$, so that $\lim |B_n(-1)|^{1/n} = \beta_B$ when $n \to \infty$ through even values of n. A similar proof shows that the same holds for odd values.

The rest of the proof is essentially unchanged, still using the functions $C_m(t)$. Since e^{2it} is of modulus 1 when t is real, the domains of analycity required for $w(\sin^2 t)$ are still the same. We thus obtain $\lim |B_{2m}(-1)R_{2m}^B|^{1/m} \le 1/r_0$ and similarly for $2m+1$, hence $\lim |B_n(-1)R_n^B|^{1/n} \le 1/\sqrt{r_0}$ as claimed.

Remarks.

(1) More generally we can consider Algorithm 2_Q with Q_n proportional to $P_n^{(m)}$ with $m = n/(1+\nu) + \mathrm{O}(1)$ and $\nu > 0$, Algorithm 2_A corresponding to $\nu = 0$ and Algorithm 2_B to $\nu = 1$ (the exact choice of m has no effect on the speed of convergence of the algorithm). The same analysis as before shows that the analogue of Proposition 2 remains true (with the *same* curves CA and CB as before), but with the numbers β and γ occurring there replaced by

$$\beta_\nu = \left(e^{\nu\alpha}\,(\cosh\alpha + \nu\sinh\alpha)\right)^{1/(\nu+1)}, \qquad \gamma_\nu = r_0^{1/(\nu+1)}\,\beta_\nu\,,$$

where $\alpha = \alpha_\nu$ is a root of $\alpha - 1/(\coth\alpha + \nu) = 2L$ and $r_0 = 2.2736\ldots$ is the same number as before.

It can be shown that $\nu = 0$, i.e. Algorithm 2_A, gives the largest value of γ_ν, and that $\nu = 1$, i.e. Algorithm 2_B, gives the largest value of β_ν, whence the choice of these two algorithms. (Note: the largest value of γ_ν is in fact obtained for $\nu = -0.123559\ldots$, but we must restrict to non-negative values of ν since otherwise more than n terms of the sequence $\{a_k\}$ are used.)

(2) We do not claim that the sequences of polynomials that we have given give the best results, only that they are natural choices. Other sequences of polynomials P_n can be used for linear acceleration of alternating sequences which for certain classes of sequences $\{a_k\}$ will give even better convergence. These sequences of polynomials are related to polynomials which are used in Diophantine approximation to get good irrationality measures for numbers such as $\log 2$, π or $\zeta(3)$, following the works of Apéry, Beukers, Rhin et al.

(3) For sequences $\{a_k\}$ for which $w(x)$ is analytic in an even larger region than that bounded by Curve CB, we can improve the 17.93^{-n} bound for Algorithm 2_A by taking a linear combination of a_k and some standard sequence like $a_k^0 = \frac{1}{k+1}$ to force $w(x)$ to vanish at -1. Then $S_n = S + w(-1)(S_n^0 - S^0) + \varepsilon_n$ where the error term ε_n tends to 0 faster than before (and even more than exponentially if $w(x)$ is entire, since the modified series $\sum Q_n(-1)R_n u^n = \int_0^{\pi/2}$ (entire in t, u) dt has infinite radius of convergence), and using this with two different values of n to eliminate the $w(-1)$ term we get improved approximations to S.

Conclusion.

Since we have given three algorithms, some advice is necessary to be able to choose between the three.

If the sequence $\{a_k\}$ is very easy to compute and not too many decimal digits are required (say at most 1000), then we suggest using Algorithm 1, which has the advantage of being by far the simplest to implement and which does not require any storage. This is the default choice made in the Pari system for example.

If the sequence $\{a_k\}$ already converges to 0 at a geometric rate, then $w(x)$ cannot be analytic and hence Algorithm 1 should again be chosen.

If the sequence $\{a_k\}$ is difficult to compute or if a large number of decimal digits are desired, it should be better to use Algorithms 2_A or 2_B. Since a priori one does not know the analytic behavior of $w(x)$, in view of the examples which have been presented, $w(x)$ has frequently a singularity at $x = 0$, hence we suggest using Algorithm 2_B. Of course, if we know that $w(x)$ is much better behaved, then Algorithm 2_A becomes useful also.

The above algorithms can also be used in cases where alternating series occur only indirectly.

A first example is the summation of series with *positive* terms. Using a trick due to Van Wijngaarden and described in [PFTV], such a series can be converted to an alternating series as follows:

$$\sum_{k=1}^{\infty} a_k = \sum_{m=1}^{\infty} (-1)^m b_m \quad \text{with} \quad b_m = \sum_{k=0}^{\infty} 2^k a_{2^k m}\,.$$

In this case the coefficients b_m of the alternating sum are themselves infinite sums, hence are hard to compute, and so it is usually useful to use Algorithm 2_B instead of Algorithm 1.

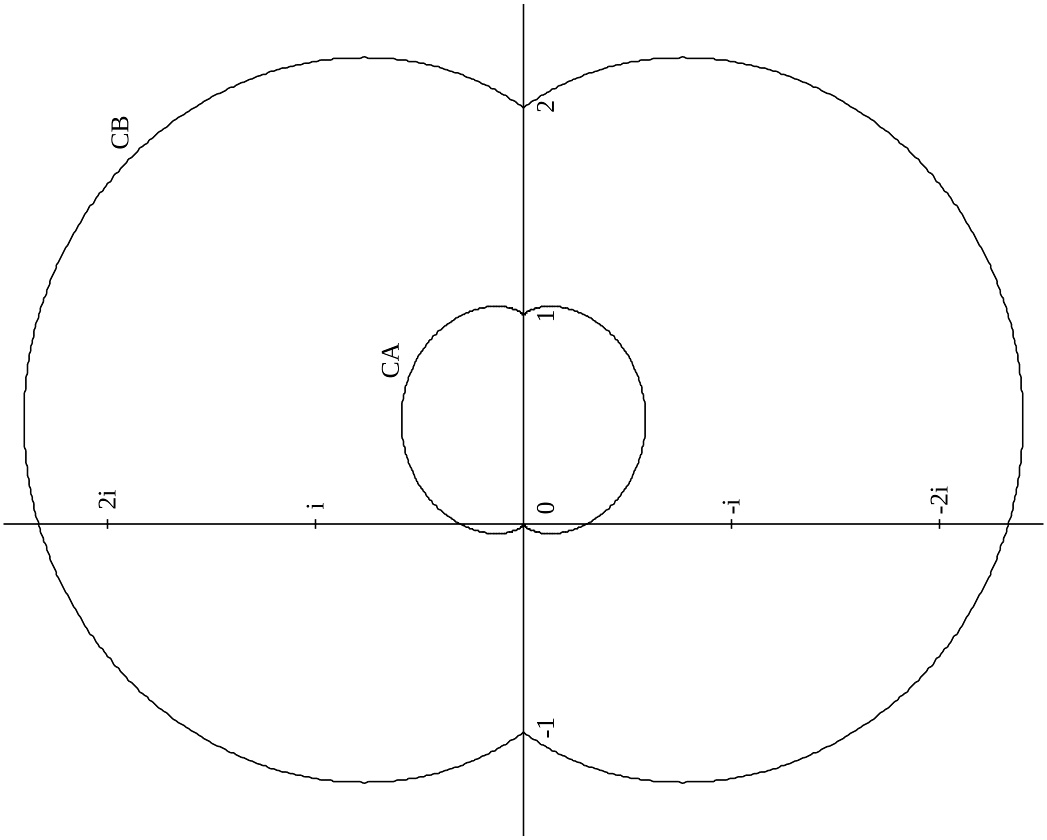
A second example is the computation of continued fractions $S = b_1/(c_1 + b_2/(c_2 + \cdots))$ with b_k and c_k positive. The standard theory of continued fractions shows that one can rewrite S as an "alternating" sum

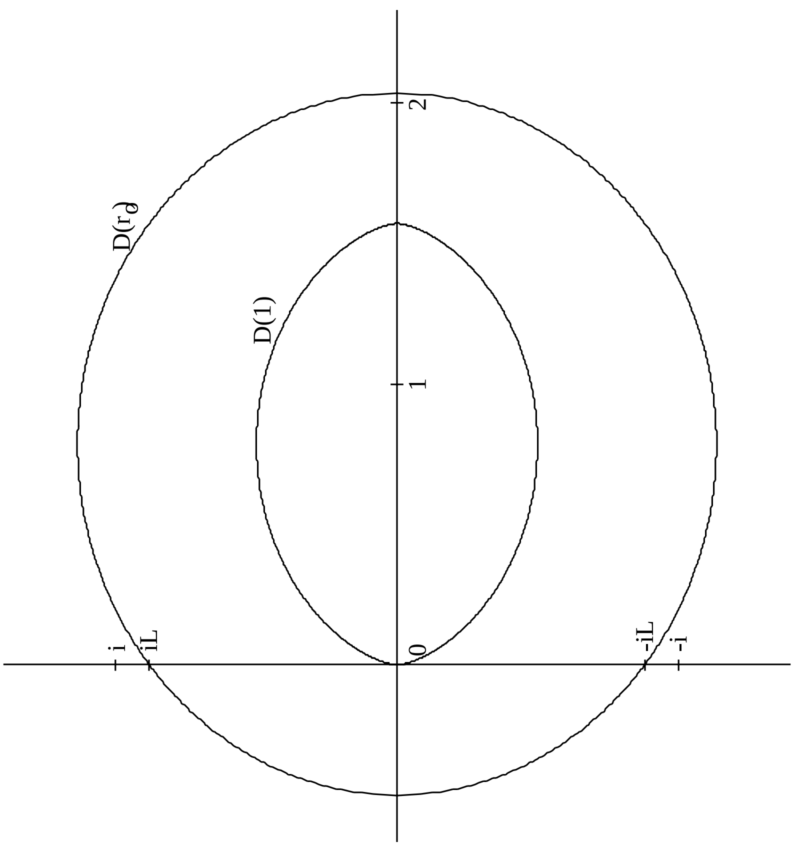
$$S = \frac{b_1}{q_0 q_1} - \frac{b_1 b_2}{q_1 q_2} + \frac{b_1 b_2 b_3}{q_2 q_3} - \cdots$$

where the q_i are defined by $q_{-1} = 0$, $q_0 = 1$ and $q_n = c_n q_{n-1} + b_n q_{n-2}$, and we can then apply one of the above algorithms to the sequence

$$a_k = \frac{\prod_{1 \le j \le k+1} b_j}{q_k q_{k+1}}\,.$$

Note that frequently continued fractions converge geometrically (this is true for example in the case of *simple* continued fractions, i.e. $b_k = 1$ for all k and a_k positive integers) hence Remark 8 after Algorithm 1 must be taken into account, and Algorithm 1 should usually be preferred.


CB
CA
2i
i
0
-i
-2i
2
1
-1


D(r)
D(1)
i
iL
0
-iL
-i
2
1

References

[BC] J. Borwein and R. Crandall, in preparation

[B] C. Brezinski, *Padé-type Approximation and General Orthogonal Polynomials*, International Series of Numerical Analysis **50**, Birkhäuser, Basel-Boston-Stuttgart (1980).

[E] M. Eiermann, *On the convergence of Padé-type approximants to analytic functions*, J. Comput. Appl. Math. **10** (1984) 219–227

[G] S.-Å. Gustafson, *Convergence acceleration on a general class of power series*, Computing **21** (1978) 53–69

[H] F. Hausdorff, *Momentprobleme für ein endliches Intervall*, Math. Z. **16** (1923) 220–248

[PFTV] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes in C* (2nd ed.), Cambridge University Press, Cambridge (1988).

[Sz] G. Szegö, *Orthogonal Polynomials*, AMS Colloquium Publications **23** (AMS, Providence, 1939)

[W] J. Wimp, *Sequence Transformations and their Applications*, Mathematics in Science and Engineering **154** (Academic Press, New York, 1981)

Max Planck Institut für Mathematik, Gottfried Claren Strasse 26, 53225 BONN, Germany