Un algorithme efficace pour la fonction ζ de Riemann

P. Borwein[1]

Cet article a été écrit à l'occasion de la thèse honoraire du Dr. Jonathan M. Borwein.

Résumé. Une classe très simple d'algorithmes pour le calcul de la fonction ζ de Riemann en précision arbitraire dans des domaines arbitraires est proposée. Ces algorithmes sont en concurrence avec les méthodes basées sur la formule de sommation d'Euler-Maclaurin, ils sont plus faciles à implémenter et ils sont plus faciles à analyser.

1. Introduction

Nous proposons quelques algorithmes très simples pour le calcul en précision arbitraire de la fonction ζ de Riemann qui est le prolongement analytique de

$$\zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s} \qquad \mathfrak{Re}(s) > 1 \tag{1.1}$$

$$= \frac{1}{1-2^{1-s}} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^s} \qquad \mathfrak{Re}(s) > 0$$

Ces algorithmes ne concurrencent pas les algorithmes basés sur la formule de Riemann-Siegel pour effectuer des calculs concernant les zéros sur la droite critique ($\mathfrak{Im}(s) = 1/2$) dans lesquels des évaluations en multiple précision sont nécessitées (voir 2,5]). Ils peuvent cependant améliorer les algorithmes standard pour des calculs en précision arbitraire de la fonction ζ, de la plupart des bibliothèques principales d'algèbre symbolique (toutes les bibliothèques de Maple, Mathematica et Pari utilisent des algorithmes basés sur la formule d'Euler-Maclaurin [2.5,7]). Ils sont plus faciles à implémenter et beaucoup plus faciles à analyser.

2. Algorithmes

On commence par présenter l'algorithme sous forme générique et on en fournit ensuite deux spécialisations.

Algorithme 1. Appelons $p_n(x) := \sum_{k=0}^{n} a_k x^k$ un polynôme arbitraire de degré n qui ne s'annule pas en -1. Appelons

$$c_j := (-1)^j \left(\sum_{k=0}^{j} (-1)^k a_k - p_n(-1) \right) \tag{2.1}$$

Alors

Transcription, traduction : Denise Vella-Chemla, novembre 2024.

Référence : http://www.cecm.sfu.ca/ pborwein/PAPERS/P155.pdf.

[1]Société canadienne de mathématiques, Proceedings de conférence.

$$(2.2) \qquad \zeta(s) = \frac{-1}{(1-2^{1-s})p_n(-1)} \sum_{j=0}^{n-1} \frac{c_j}{(1+j)^s} + \xi_n(s)$$

où

$$(2.3) \qquad \xi_n(s) = \frac{1}{p_n(-1)(1-2^{1-s})} \frac{1}{\Gamma(s)} \int_0^1 \frac{p_n(x)|\log x|^{s-1}}{1+x} dx.$$

Ici Γ est la fonction Gamma.

Notons que les c_j sont juste (au signe près) les coefficients de $\frac{p_n(x) - p_n(-1)}{1+x}$ qui est un polynôme de degré $n-1$.

Preuve. On utilise les formules standard

$$(2.4) \qquad \zeta(s) = \frac{1}{(1-2^{1-s})\Gamma(s)} \int_0^1 \frac{|\log x|^{s-1}}{1+x} dx \qquad \mathfrak{Re}(s) > 0$$

et

$$(2.5) \qquad \frac{1}{(m+1)^s} = \frac{1}{\Gamma(s)} \int_0^1 x^m |\log x|^{s-1} dx \qquad \mathfrak{Re}(s) > 0$$

Voir [1,7], cependant les deux découlent facilement de

$$(2.6) \qquad \Gamma(s) = \int_0^\infty u^{s-1} e^{-u} du = \int_0^1 |\log x|^{s-1} dx \qquad \mathfrak{Re}(s) > 0$$

qui est juste la définition de Γ et (1).

On écrit maintenant

$$\begin{aligned}
\xi_n(s) : \; &= \frac{1}{p_n(-1)(1-2^{1-s})} \frac{1}{\Gamma(s)} \int_0^1 \frac{p_n(x)|\log x|^{s-1}}{1+x} dx \\
&= \frac{1}{p_n(-1)(1-2^{1-s})} \frac{1}{\Gamma(s)} \int_0^1 \frac{p_n(-1)|\log x|^{s-1}}{1+x} dx \\
&= \frac{1}{p_n(-1)(1-2^{1-s})} \int_0^1 \frac{p_n(-1) - p_n(x)}{1+x} |\log x|^{s-1} dx.
\end{aligned}$$

Le premier terme ci-dessus donne $\zeta(s)$ par (2.4) et le dernier terme se développe avec (2.5) pour donner le développement en série dans (2.2). $\square$

La ruse maintenant est de choisir p_n de telle façon que l'erreur dans l'intégrale pour ξ_n divisée par $p_n(-1)$ soit aussi petite que possible.

Le polynôme de Chebychev, décalé vers $[0,1]$, et convenablement normalisé, maximise la valeur $p_n(-1)$ sur tous les polynômes de norme supremum comparable sur $[0,1]$. Donc les polynômes de Chebychev sont le choix évident pour p_n et ils fournissent le prochain résultat.

Algorithme 2. Soit

$$d_k := n \sum_{i=0}^{k} \frac{(n+i-1)!4^i}{(n-i)!(2i)!}$$

alors

$$\zeta(s) = \frac{-1}{d_n(1-2^{1-s})} \sum_{k=0}^{n-1} \frac{(-1)^k(d_k - d_n)}{(k+1)^s} + \gamma_n(s)$$

où pour $s = \sigma + it$ avec $\sigma \geq \frac{1}{2}$

$$\begin{aligned} |\gamma_n(s)| &\leq \frac{2}{(3+\sqrt{8})^n)} \frac{1}{|\Gamma(s)|} \frac{1}{|(1-2^{1-s})|} \\ &\leq \frac{3}{(3+\sqrt{8})^n} \frac{(1+2|t|)e^{\frac{|t|n}{2}}}{|(1-2^{1-s})|} \end{aligned}$$

Preuve. La formule dont on a besoin pour le n-ième polynôme de Chebychev sur $[0,1]$ est

$$T_n(x) = (-1)^n n \sum_{k=0}^{n} (-1)^k \frac{(n+k-1)!}{(n-k)!(2k)!} 4^k x^k$$

dont l'expression pour d_k se déduit. Pour estimer l'erreur, on observe que, par l'algorithme 1,

$$\begin{aligned} |\gamma_n(s)| &= \left| \frac{1}{d_n(1-2^{1-s})} \frac{1}{\Gamma(s)} \int_0^1 \frac{T_n(x)|\log(x)|^{s-1}}{1+x} dx \right| \\ &\leq \frac{2}{(3+\sqrt{8})^n} \frac{1}{|(1-2^{1-s})\Gamma(s)|} \int_0^1 \frac{|\log x|^{s-1}}{1+x} dx \end{aligned}$$

puisque sur $[0,1]$, $|T_n(x)|$ est borné par 1 et $|T_n(-1)| \geq \frac{1}{2}(3+\sqrt{8})^n$. On calcule maintenant que

$$\int_0^1 \frac{|\log x|^{\frac{1}{2}}}{1+x} dx \leq .68$$

pour en déduire que

$$|\gamma_n(s)| \leq \frac{1.36}{(3+\sqrt{8})^n} \frac{1}{|(1-2^{1-s})\Gamma(s)|}$$

Maintenant, pour $s = \sigma + it$ avec $\sigma \geq \frac{1}{2}$

$$\left| \frac{\Gamma(\sigma)}{\Gamma(\sigma+it)} \right|^2 = \prod_{n=0}^{\infty} \left((1 + \frac{t^2}{(\sigma+n)^2} \right).$$

Donc

$$\frac{1}{|\Gamma(s)|} = \frac{\left(\prod_{n=0}^{\infty}(1+\frac{t^2}{(\sigma+n)^2})\right)^{1/2}}{|\Gamma(\sigma)|}$$

$$\leq \frac{\left(\prod_{n=0}^{\infty}(1+\frac{t^2}{(\frac{1}{2}+n)^2})\right)^{1/2}}{|\Gamma(\sigma)|}$$

$$\leq \frac{\left(\frac{1+4t^2}{|t|\pi}\right)^{\frac{1}{2}}(\sinh(t\pi))^{\frac{1}{2}}}{|\Gamma(\sigma)|}$$

Puisque $|\Gamma(\sigma)|^{-1} \leq 1.5$ sur $[\frac{1}{2},\infty)$, on est parvenu au but. □

Puisque $(3+\sqrt{8}) = 5.828\ldots$, et que ceci est le terme dominant dans l'estimation, on voit qu'on a besoin d'environ $(1.3)\pi$ termes pour n chiffres de précision, en supposant qu'on est près de l'axe réel.

Un algorithme encore plus simple, bien qu'il ne soit pas aussi rapide, peut être basé sur le choix de $p_n(x) := x^n(1-x)^n$.

Algorithme 3. Soit

$$e_j = (-1)^j \left[\sum_{k=0}^{j-n} \frac{n!}{k!(n-k)!} - 2^n\right]$$

(où la somme vide est nulle). Alors

$$\zeta(s) = \frac{-1}{2^n(1-2^{1-s})}\sum_{j=0}^{2n-1}\frac{e_j}{(j+1)^s} + \gamma_n(s)$$

où, pour $s = \sigma + it$ avec $\sigma > 0$

$$|\gamma_n(s)| \leq \frac{1}{8^n}\frac{(1+|\frac{t}{\sigma}|)e^{\frac{|t|\pi}{2}}}{|1-2^{1-s}|}$$

Si $-(n-1) \leq \sigma < 0$ alors

$$|\gamma_n(s)| \leq \frac{1}{8^n|1-2^{1-s}|}\frac{4^{|\sigma|}}{|\Gamma(s)|}.$$

(Notons que les $\gamma_n(s) = 0$ pour $s = -1, -2, ..., -n+1$.)

Les détails de cela sont très similaires à ceux de l'algorithme 2 sur le fait d'utiliser $p_n(x) := x^n(1-x)^n$ et on les omet. Le fait que la convergence persiste dans la partie du demi-plan $\{\mathfrak{Re}(s) < 0\}$ est une conséquence du fait que

$$\int_0^1 \frac{x^n(1-x)^n}{1+x}|\log x|^{s-1}dx$$

converge si la contrainte $\mathfrak{Re}(s) > -n$ est respectée. Ainsi, l'algorithme 3 donne une autre preuve du prolongement analytique de $\zeta(s)(1-s)$. (Notons que $|\epsilon_j/2^n| = 1$ pour $j = 0, \ldots, n$ et $|e_j/2^n| \leq 1$ pour tout j.)

Parce que $1/\Gamma(s) = 0$ pour s un entier négatif, on a que $\gamma_n(s) = 0$ pour $s = -1, -2, ..., -n+1$. Pourtant, puisque

$$\zeta(-2n+1) = -\frac{\beta_{2n}}{2n}$$

la somme dans l'algorithme 2 calcule les nombres de Bernoulli, pour $s = -1, ..., -n+1$ exactement.

Pour faire des comparaisons, un certain soin doit être pris. Pour un calcul basé sur la formule d'Euler-Maclaurin, les nombres de Bernoulli doivent être calculés. S'ils sont alors stockés, un second calcul sera beaucoup plus rapide que l'évaluation initiale. Une partie de ce qui rend la formule d'Euler-Maclaurin rebutante pour les calculs en très grande précision est qu'elle a besoin de beaucoup de mémoire et qu'il est coûteux en terme de calculs de calculer les nombres de Bernoulli, au moins par les méthodes habituelles. Pour le dire rapidement, il faut calculer les nombres de Bernoulli jusqu'à l'ordre n environ pour obtenir une précision à n chiffres et cela nécessite un surcroît de mémoire de l'ordre de n^2.

Les coefficients dans le genre des coefficients binomiaux des algorithmes 2 et 3 sont beaucoup plus faciles à calculer et si on les exécute séquentiellement, cela nécessite seulement un coefficient binomial supplémentaire par terme, ce qui se calcule par une seule multiplication et une seule division.

3. Optimalité

Les algorithmes 2 et 3 sont presque optimaux au sens suivant. Il n'y a pas de séquence de polynômes exponentiels de n-termes qui peuvent converger vers $\zeta(s)$ sur un intervalle $[a, b], a > 1$ beaucoup plus vite que ne le font les polynômes des algorithmes proposés. Plus précisément, on a le

Théorème 3.1. *Soit* $1 < \alpha < \beta$ *et soit* n *fixé. Alors*

$$\left\| \zeta(s) - \sum k = 1^n \frac{a_k}{b_k^s} \right\|_{[\alpha,\infty)} \geq \frac{1}{(2^\alpha(3+\sqrt{8})^2)^n}$$

et

$$\left\| \zeta(s) - \sum_{k=1}^{n} \frac{a_k}{b_k^s} \right\|_{[\alpha,\beta)} \geq (D(\alpha,\beta))^n$$

pour tous réels (a_k) *et* (b_k). *Ici* $D(\alpha, \beta)$ *est une constante positive qui ne dépend que de* α *et* β *et* $\|.\|_{[\alpha,\beta]}$ *désigne la norme supremum sur* $[\alpha, \beta]$.

Preuve. La preuve suit la méthode de [4]. Selon le changement de variables $s \to -\log x / \log 2$ pour un certain réel $(c_k), (d_k)$ et (e_k)

$$\left\|\zeta(s)-\sum_{k=1}^{n}\frac{a_k}{b_k^s}\right\|_{[\alpha,\beta]}$$

$$=\left\|\sum_{k=1}^{\infty}x^{\log k/\log 2}-\sum_{k=1}^{n}a_kx^{c_k}\right\|_{[2^{-\beta},2^{-\alpha}]}$$

$$\geq\left\|\sum_{k=0}^{n}x^{k}-\sum_{k=1}^{n}d_kx^{e_k}\right\|_{[2^{-\beta},2^{-\alpha}]}$$

où la dernière inégalité découle du théorème de comparaison (Corollaire 2 de [4]). Maintenant, le théorème 8 de [4] fournit l'estimation explicite

$$\left\|\sum_{k=0}^{n}x^{k}-\sum_{k=1}^{n}d_kx^{e_k}\right\|_{[2^{-(\beta-\alpha)},1]}\geq\frac{1}{(C+\sqrt{C^2-1})^{2n}}$$

où $C := (3+2^{-(\beta-\alpha)})/(1-2^{-(\beta-\alpha)})$ et le résultat en découle avec l'aide du corollaire 2 de 4 à nouveau. $\square$

Une autre manière selon laquelle les algorithmes 2 et 3 sont (en quelque sorte) presque optimaux est la suivante. Aux entiers pairs, les algorithmes génèrent des approximations rationnelles qui satisfont, pour tout entier positif N,

$$\left\|\zeta(2N)-\frac{p_n}{q_n}\right\|<\frac{1}{q_n^{\epsilon}}$$

pour une infinité d'entiers $(p_n),(q_n)$ et quelques entiers positifs $\epsilon := \epsilon(N)$. Mais les résultats de Mahler montre qu'une telle inégalité existe avec ϵ arbitrairement grand et on s'attend à ce qu'en fait, ϵ ne puisse pas être plus grand que 2 (voir le chapitre 11 de 3).)

Références

[1] M. Abromowitz, I. Stegun, *Handbook of Mathematical Functions*, Dover, N.Y., 1972.

[2] M. V. Berry, J. P. Keating. *A new asymptotic representation for $\zeta(\frac{1}{2}+it)$ and quantum spectral determinants*, Proc. R. Soc. Lond. A. 437 (1992), 151-173.

[3] J. Borwein, P. Borwein, *Pi and the AGM*, Wiley N.Y., 1987.

[4 P. Borwein, *Uniform approximation by polynomials with variable exponents*, Canadian J. Math. 35 (1983), 547-557.

[5] H. Cohen, M. Olivier, *Calcul des valeurs de la function zêta de Riemann en multiprécision*, C.R. Acad. Sci. Paris, Série 1 314 (1992), 427-430.

[6] A. Odlyzko, A. Schönhage. *Fast algorithms for multiple evaluations of the Riemann zeta-function.* Trans A.M.S. 309 (1988), 797-809.

[7] E. C. Titchmarsh, *The Theory of the Riemann Zeta-function*, Second ed. Oxford Scientific Publication, 1986.

Département de mathématiques et statistiques, Université Simon Fraser, Burnaby, BC V5A 186, Canada.