Évaluation à grande précision de $\Gamma(p/q)$

Encadrant : Marc Mezzarobba
Nombre d'étudiants : 2

La fonction gamma est une fonction spéciale classique définie sur $\mathbb{C}\backslash\mathbb{Z}_-$ qui satisfait

$$\Gamma(n+1) = n! \qquad \text{pour } n \in \mathbb{N}.$$

C'est donc une forme de prolongement de la factorielle aux nombres complexes.

On ne connaît pas d'algorithme pour évaluer numériquement $\Gamma(z)$ à précision d en temps $o(d^{3/2})$ pour un nombre complexe z quelconque. Cependant, R. Brent [1, 2] a observé que quand $z = p/q$ est un rationnel fixé, il est possible de calculer $\Gamma(p/q)$ à 2^{-d} près en $O(d\,(\log d)^{O(1)})$ opérations. Une façon de faire s'appuie sur l'écriture

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{e^{-t}\,t^{z+n}}{z\,(z+1)\cdots(z+n)} + \int_t^{\infty} e^{-u}\,u^{z-1}\,\mathrm{d}u,$$

où, pour t bien choisi, l'intégrale est petite et les sommes partielles de la série peuvent être calculées très rapidement par une technique simple appelée *scindage binaire*. En pratique, cette méthode est intéressante quand le dénominateur q est assez petit par rapport à la précision de calcul.

Dans le cadre de ce projet, vous devrez :

- étudier l'algorithme mentionné ci-dessus, son analyse de complexité et son analyse d'erreur, à partir notamment de l'article de B. Haible et T. Papanikolaou [3],
- l'implémenter en langage C en utilisant les bibliothèques FLINT [4] et Arb [5],
- évaluer expérimentalement les performances de votre implémentation et la comparer à celle présente dans Arb[1],
- améliorer votre version (éventuellement en vous inspirant de l'article de F. Johansson [6] qui contient une description de l'implémentation dans Arb, ou en réutilisant certaines sous-routines) pour essayer de faire mieux dans le cas des points rationnels de petit déonominateur,
- si le temps le permet, généraliser le travail précédent au cas des points $z \in \mathbb{C}$ racines de polynôme à « petits » coefficients entiers, qui peut se faire en utilisant la même idée.

Bibliographie

[1] Richard P. Brent. The complexity of multiple-precision arithmetic. In R. S. Anderssen and R. P. Brent, editors, *The Complexity of Computational Problem Solving*, pages 126–165, Brisbane, Australia, 1976. URL: `http://wwwmaths.anu.edu.au/~brent/pub/pub032.html`.

[2] Richard P. Brent. A Fortran multiple-precision arithmetic package. *ACM Transactions on Mathematical Software*, 4(1):57–70, March 1978. URL: `http://maths.anu.edu.au/~brent/pub/pub042.html`.

[3] Bruno Haible and Thomas Papanikolaou. Fast multiprecision evaluation of series of rational numbers, 1997. URL: `http://www.ginac.de/CLN/binsplit.pdf`.

[4] William Hart, Sebastian Pancratz, Andrew Novocin, Fredrik Johansson, David Harvey, et al. Flint: Fast library for number theory, 2007-. URL: `http://flintlib.org/`.

[5] Fredrik Johansson. Arb, 2012-. URL: `http://fredrikj.net/arb/`.

[6] Fredrik Johansson. Evaluating parametric holonomic sequences using rectangular splitting. In Nabeshima [7]. URL: `http://fredrikj.net/math/rectangular_splitting.pdf`.

[7] Katsusuke Nabeshima, editor. *ISSAC 2014: Proceedings of the 39th international symposium on Symbolic and algebraic computation*. ACM, 2014.

1. L'implémentation de Γ dans Arb est la plus rapide qui existe dans le cas général, mais ne tire pas parti du cas particulier des points rationnels, sauf pour les tous petits dénominateurs.