Offre de thèse

Julien Clément Matthieu Dien Martin Pépin

23 avril 2026

1 Informations générales

Intitulé Boltzmann Uniform Generation for Differential Equations Languages (BoUGDELa).

Établissement Université de Caen Normandie.

Laboratoire Groupe de Recherche en Informatique, Image et Instrumentation (GREYC).

Période de la thèse Octobre 2026 — Septembre 2029

2 Contexte et objectifs

La génération aléatoire a de nombreuses applications dans des domaines scientifiques variés :

- pour la simulation et la compréhension de phénomènes physiques, par exemples
  - le modèle d'Ising permettant de décrire les matériaux ferromagnétiques [Vel09],
  - les diagrammes de cordes et cartes combinatoires liés aux diagrammes de Feynmann utilisés en théorie quantique des champs [CYZ16] ;
- en biologie, pour modéliser et étudier les structures secondaires de l'ARN [Pon06] ;
- en ingénierie logicielle pour les tests automatisés [CH00], ou le fuzzing [Edd20] ;
- en algorithmique, comme procédure pour calculer le volume de polytope [LV06], calculer le permanent de matrices [JS89], etc ;
- en informatique mathématique comme outils d'expérimentations [BM22] et d'interprétations [Bet].

Évidemment, ces champs d'applications variés n'ont pas tous les mêmes besoins algorithmiques : dans certains cas la distribution des objets générés doit parfaitement être contrôlée, dans d'autres la performance de l'algorithme de génération est primordiale, ou encore, l'algorithme doit être générique pour permettre la génération de différents types d'objets. Les méthodes ad-hoc tirent parti de propriétés intrinsèques aux objets à générer ou à la distribution à échantillonner. Par exemple, l'algorithme de Rémy [Rém85] pour la génération aléatoire des arbres binaires, a une complexité temporelle linéaire en la taille des objets générés. En général, ces algorithmes sont très efficaces mais difficilement adaptables.

Les méthodes génériques fonctionnent à partir d'une description des objets à générer. Par exemple, le couplage depuis le passé [PW98] donne une méthodologie pour échantillonner une distribution décrite par une chaîne de Markov ergodique (sous certaines conditions). De même, pour les objets définis de manière inductive, la méthode récursive [NW78] permet de générer aléatoirement ces objets, uniformément, parmi les objets de même taille (fixée).

Parmi ces méthodes génériques, la méthode de Boltzmann, introduite par [Duc+04] se distingue par sa généricité et son efficacité.

Une grande part des structures discrètes peut-être définie de manière inductive, par exemple : la structure secondaire de l'ARN, les diagrammes de cordes, les structures de données, etc. Le langage des spécifications combinatoires [FS09] est un formalisme très flexible permettant de décrire ces familles de structures. Par exemple, un arbre binaire $\mathcal{T}$ est soit une feuille $\square$ soit un nœud $\bullet$ et une paire de descendants (qui sont aussi des arbres binaires) $\mathcal{T} \times \mathcal{T}$. Ainsi, la classe combinatoire des arbres binaires a pour spécification

$$\mathcal{T} = \square \cup \bullet \times \mathcal{T} \times \mathcal{T}.$$

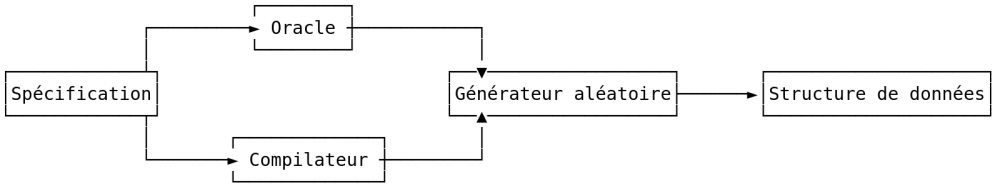

FIGURE 1 – Schéma de la méthode de Boltzmann

À partir de cette spécification on peut compiler un générateur aléatoire de structures. Ce générateur produit des structures selon la distribution de Boltzmann qui a les propriétés de :

— se composer simplement selon les opérateurs des spécifications, par exemple : la distribution de Boltzmann $\Gamma[\mathcal{A} \times \mathcal{B}]$ du produit cartésien des classes $\mathcal{A}$ et $\mathcal{B}$ est le produit des distributions de Boltzmann pour $\mathcal{A}$ et $\mathcal{B}$, c'est-à-dire $\Gamma[\mathcal{A} \times \mathcal{B}] = \Gamma[\mathcal{A}] \otimes \Gamma[\mathcal{B}]$;

— les objets de même taille ont la même probabilité d'être générés, par exemple avec un générateur de Boltzmann pour les arbres binaires, les arbres $\bullet(\bullet(\square, \square), \bullet(\square, \square))$ et $\bullet(\bullet(\square, \bullet(\square, \square)), \square)$ auront la même probabilité d'êtres tirés. Ainsi, la taille des objets générés n'est pas fixée mais est, elle aussi, aléatoire.

Pour viser une taille, un certain nombre de paramètres doivent, selon la spécification, être calculés. C'est le rôle de l'oracle. Dans les cas « simples », l'oracle évalue numériquement les fonctions génératrices obtenues, elles aussi, à partir de la spécification.

L'objectif de cette thèse est d'enrichir cette méthode sur le plan théorique et sur le plan pratique.

3 Projet détaillé

3.1 Verrous abordés et travail scientifique

Depuis l'introduction de la méthode en 2004, un certain nombre de développements théoriques ont eu lieu autour de la génération de Boltzmann, selon deux grands axes.

D'une part, l'expressivité du langage de spécification, permettant de décrire les langages à générer, a été augmentée en étudiant de nouvelles constructions combinatoires. Cela a permis en particulier d'intégrer la prise en charge des structures étiquetées croissantes [BRS12 ; Bod+16] ainsi que d'un certain nombre de structures non étiquetées [FFP07] en prenant en compte leurs symétries. D'autre part, la question de l'implémentation efficace et précise des oracles a été étudiée d'un point de vue algorithmique et calcul formel, ce qui a donné lieu à deux approches différentes pour leur implémentation : l'une basée sur l'itération de Newton [PSS08] et l'autre basée sur l'optimisation convexe [BBD18]. Cependant, un certain nombre de verrous théoriques demeurent.

En particulier, les constructions combinatoires donnant lieu à des systèmes différentiels ne sont que très partiellement prises en charge, bien que très présentes en combinatoire. Par exemple, l'opération de pointage (sélection d'un atome arbitraire dans la structure) donne lieu à des familles de structures pour lesquelles la méthode de Boltzmann ne s'applique pas de façon automatique. Un des objectifs de la thèse sera d'**étendre la théorie pour pouvoir gérer le pointage**. Un cas typique d'application du pointage est la spécification des 1-2-arbres [CDD25].

Un second enjeu est de comprendre l'impact des erreurs numériques de l'oracle sur la distribution de sortie d'un générateur de Boltzmann. La correction des algorithmes repose sur une arithmétique exacte pour pourvoir générer des objets selon le modèle de Boltzmann. Cependant, les implémentations existantes reposent sur de l'arithmétique en virgule flottante. Une hypothèse communément admise est que les erreurs introduites par l'utilisation de nombres flottants ont un impact négligeable sur le résultat du générateur. Un second objectif de la thèse sera de **quantifier l'impact réel de ces erreurs** à l'aide d'une analyse rigoureuse de leur propagation.

Enfin, deux approches émergent comme contournement possible aux erreurs numériques. L'une de ces approches repose sur une adaptation des générateurs pour introduire un rejet venant corriger le biais théorique [BLR15]. Cette approche n'est applicable que pour un ensemble restreint de constructions et ne semble pas avoir été implémentée dans les librairies existantes. L'autre approche consiste à implémenter des oracles paresseux, capables d'affiner leur précision dynamiquement en fonction des besoins du générateur à l'exécution. Cette seconde approche a l'avantage d'être applicable très largement. Un troisième objectif de cette thèse sera de **formaliser, implémenter et comparer les oracles paresseux aux approches existantes**.

Tous les développements algorithmiques de cette thèse seront accompagnés d'une implémentation dans la bibliothèque `usainboltz`, développée à l'Université de Caen dans l'équipe Amacc, et sujets à une intégration dans le logiciel Sagemath (logiciel libre de calcul mathématiques largement diffusé). La librairie `usainboltz` est accessible à l'adresse `https://gitlab.com/ParComb/usain-boltz` est distribuée sous licence (libre) GPLv3.0. L'intégration dans Sagemath se fera en implémentant directement les outils liés à l'oracle dans l'écosystème de Sagemath ainsi qu'en développant une interface entre les classes pré-existantes pour la combinatoire et les grammaires `usainboltz`.

4 Principales actions et calendrier détaillé de mise en œuvre

Les travaux de thèse se dérouleront selon le calendrier (prévisionnel) suivant :

T00-T06. La personne recrutée pour la thèse devra établir un état de l'art afin de se familiariser avec les constructions connues et les implémentations existantes. De plus, l'implémentation d'un générateur exhaustif des structures (de petite taille) décrites par une spécification combinatoire est attendue. Ce travail d'implémentation vient remplir trois objectifs :

- se familiariser avec les outils théoriques et la notion de spécification combinatoire ;
- se familiariser avec la base de code existante ;
- ajouter à `usainboltz`] une fonctionnalité (génération exhaustive de petites structures) dont le besoin a déjà été identifié dans la communauté.

Ces travaux donneront lieu à une montée de version dans `usainboltz`.

T06-T12. Dans un second temps, la personne recrutée devra implémenter l'approche décrite dans les articles [BRS12] et [Bod+16] pour les étiquetages contraints. Cela servira de travail préparatoire pour développer une nouvelle approche, plus générique, pour prendre en charge l'opérateur de pointage. En termes de compilation, cela implique de développer un nouvel algorithme implémentant la distribution de Boltzmann efficacement pour cet opérateur. Concernant l'oracle, dans le cas particulier D-fini, nous pourrons notamment nous appuyer sur les travaux de Marc Mezzarobba [BJM17]. Ce travail pourra être valorisé par la soumission d'un article à une conférence.

T12-T18. La personne recrutée devra intégrer les méthodes numériques de l'état de l'art [PSS08 ; BJM17 ; BBD18] dans `usainboltz`. Il s'agira dans un premier temps de prendre en mains ces outils théoriques, puis dans un second temps de les adapter pour une utilisation dans un cadre paresseux. Ce travail sera valorisé par des contributions dans le logiciel sous licence libre Sagemath afin de rendre ces résultats facilement accessibles à la communauté scientifique.

T18-T24. La personne recrutée proposera une étude théorique et expérimentale de l'influence de la précision de l'oracle sur la distribution de sortie des générateurs de Boltzmann. Ce travail permettra de déterminer si les erreurs numériques ont un impact réel en pratique et surtout dans quel régime de taille. Nous souhaitons également identifier les situations où l'approche paresseuse et l'approche développée dans [BLR15] deviennent nécessaires. Ces résultats seront valorisés par l'écriture d'un second article scientifique.

T24-T30. La thèse proposée s'articule autour de l'ajout de l'opérateur de pointage et des outils algorithmiques qui l'accompagnent au niveau de l'oracle. Une perspective naturelle de cette thèse, qui sera développée lors de la troisième année, est l'extension de la méthode à l'opérateur de différence finie. Cet opérateur permet de spécifier un certain nombre de structures présentant des contraintes « au bord », telles que les marches confinées dans une partie du plan ou certaines familles de cartes. Cet

opérateur a déjà été étudié dans un cadre combinatoire [BJ06] et présente des propriétés communes avec l'opérateur de pointage [BJM17 ; KJJ15]. La personne recrutée développera les outils algorithmiques manquants pour intégrer cet opérateur dans la méthode de Boltzmann, ce qui donnera lieu à l'écriture d'un troisième article scientifique.

T30-T36. Rédaction du manuscrit de thèse.

Références

[BBD18] Maciej Bendkowski, Olivier Bodini et Sergey Dovgal. « Polynomial Tuning of Multiparametric Combinatorial Samplers ». In : *2018 Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*. SIAM, 2018, p. 92-106.

[Bet] Jérémie Bettinelli. *Page Professionnelle de Jérémie Bettinnelli*.

[BJ06] Mireille Bousquet-Mélou et Arnaud Jehanne. « Polynomial Equations with One Catalytic Variable, Algebraic Series and Map Enumeration ». In : *Journal of Combinatorial Theory, Series B* 96.5 (2006), p. 623-672.

[BJM17] Alexandre Benoit, Mioara Joldeș et Marc Mezzarobba. « Rigorous Uniform Approximation of D-finite Functions Using Chebyshev Expansions ». In : *Mathematics of Computation* 86.305 (2017), p. 1303-1341. doi : 10.1090/mcom/3135.

[BLR15] Olivier Bodini, Jérémie Lumbroso et Nicolas Rolin. « Analytic Samplers and the Combinatorial Rejection Method ». In : *2015 Proceedings of the Meeting on Analytic Algorithmics and Combinatorics (ANALCO)*. SIAM, 2015, p. 40-50. doi : 10.1137/1.9781611973761.4.

[BM22] Nicolas Bonichon et Pierre-Jean Morel. *Baxter d-Permutations and Other Pattern Avoiding Classes*. 2022. arXiv : 2202.12677 [math.CO].

[Bod+16] Olivier Bodini, Matthieu Dien, Xavier Fontaine, Antoine Genitrini et Hsien-Kuei Hwang. « Increasing Diamonds ». In : *LATIN 2016 : Theoretical Informatics*. Springer, 2016, p. 207-219.

[BRS12] Olivier Bodini, Olivier Roussel et Michèle Soria. « Boltzmann Samplers for First-Order Differential Specifications ». In : *Discrete Applied Mathematics* 160.18 (déc. 2012), p. 2563-2572. doi : 10.1016/j.dam.2012.05.022.

[CDD25] Julien Courtiel, Matthieu Dien et Paul Dorbec. « Cayley Trees and Increasing 1,2-Trees : Let's Twist ! » Mars 2025. doi : 10.37236/00000.

[CH00] Koen Claessen et John Hughes. « QuickCheck : A Lightweight Tool for Random Testing of Haskell Programs ». In : *ACM SIGPLAN International Conference on Functional Programming*. 2000. doi : 10.1145/351240.351266.

[CYZ16] Julien Courtiel, Karen Yeats et Noam Zeilberger. « Connected Chord Diagrams and Bridgeless Maps ». In : *arXiv preprint arXiv :1611.04611* (2016). arXiv : 1611.04611.

[Duc+04] Philippe Duchon, Philippe Flajolet, Guy Louchard et Gilles Schaeffer. « Boltzmann Samplers for the Random Generation of Combinatorial Structures ». In : *Combinatorics, Probability & Computing* 13.4-5 (2004), p. 577-625. doi : 10.1017/S0963548304006315.

[Edd20] Michael Eddington. *Peach Fuzzer*. 2004/2020. url : https://peachtech.gitlab.io/peach-fuzzer-community/.

[FFP07] Philippe Flajolet, Eric Fusy et Carine Pivoteau. « Boltzmann Sampling of Unlabelled Structures ». In : *Workshop on Analytic Algorithmics and Combinatorics*. New Orleans, United States, jan. 2007, p. 201-211. doi : 10.1137/1.9781611972979.5.

[FS09] Philippe Flajolet et Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009, p. I-XIII, 1-810. isbn : 978-0-521-89806-5. doi : 10.1017/CBO9780511801655.

[JS89] Mark Jerrum et Alistair Sinclair. « Approximating the Permanent ». In : *SIAM journal on computing* 18.6 (1989), p. 1149-1178.

[KJJ15] Manuel Kauers, Maximilian Jaroschek et Fredrik Johansson. « Ore Polynomials in Sage ». In : *Computer Algebra and Polynomials : Applications of Algebra and Number Theory*. Springer, 2015, p. 105-125.

[LV06] László Lovász et Santosh Vempala. « Simulated Annealing in Convex Bodies and an O*(N4) Volume Algorithm ». In : *Journal of Computer and System Sciences* 72.2 (2006), p. 392-417.

[NW78] Albert Nijenhuis et Herbert Wilf. *Combinatorial Algorithms : For Computers and Hard Calculators*. 2e éd. USA : Academic Press, Inc., 1978. isbn : 0-12-519260-6. doi : 10.1016/C2013-0-11243-3.

[Pon06] Yann Ponty. « Modélisation de Séquences Génomiques Structurées, Génération Aléatoire et Applications ». Theses. Université Paris Sud - Paris XI, nov. 2006.

[PSS08] Carine Pivoteau, Bruno Salvy et Michèle Soria. « Boltzmann Oracle for Combinatorial Systems ». In : *DMTCS Proceedings*. T. DMTCS Proceedings vol. AI, Fifth Colloquium on Mathematics and Computer Science. DMTCS Proceedings. Blaubeuren, Germany : Discrete Mathematics & Theoretical Computer Science, sept. 2008, p. 475-488. doi : 10.46298/dmtcs.3585.

[PW98] James Propp et David Wilson. « Coupling from the Past : A User's Guide ». In : *Microsurveys in discrete probability* 41 (1998), p. 181-192.

[Rém85] Jean-Luc Rémy. « Un Procédé Itératif de Dénombrement d'arbres Binaires et Son Application à Leur Génération Aléatoire ». In : *RAIRO. Informatique théorique* 19.2 (1985), p. 179-195.

[Vel09] Yvan Velenik. « Le Modèle d'Ising ». DEA. Fév. 2009.