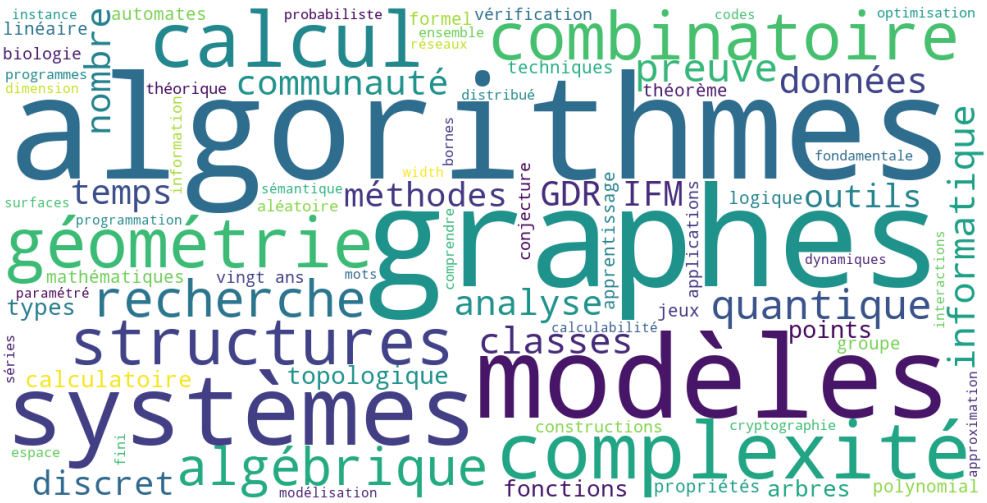
GDR Groupement de recherche
IFM Informatique Fondamentale et ses Mathématiques

Les vingt ans du GdR IFM « Informatique Fondamentale et ses Mathématiques »

17-19 mars 2026

https://20ansgdrifm.sciencesconf.org/page/panoramas



Edité par

Conseil Scientifique du GdR IFM

Disponible en ligne sur
https://20ansgdrifm.sciencesconf.org/page/panoramas

Date de publication
17 mars 2026

Le GdR IFM en quelques lignes

Les Groupements de Recherche (GdR) sont un outil du CNRS visant à animer, structurer et promouvoir les recherches scientifiques sur des thématiques spécifiques. Le GdR Informatique Fondamentale et ses Mathématiques (IFM) a été créé le 1er janvier 2024 par l'institut Sciences Informatiques du CNRS pour cinq ans. Il s'inscrit dans la suite directe du GdR Informatique Mathématique (IM) créé en 2006.

Les responsables successifs des GdR IM puis IFM ont été :

— Brigitte Vallée et Christiane Frougny (2006-2009)
— Brigitte Vallée et Arnaud Durand (2010-2013)
— Arnaud Durand et Jean-Michel Muller (2014-2018)
— Jean-Michel Muller et Guillaume Theyssier (2019-2023)
— Pierre Fraigniaud et Guillaume Theyssier (2024-2028)

Depuis sa création, le GdR couvre une très grande part des domaines fondamentaux sur lesquels repose l'informatique, et maintient à ce titre des liens très étroits avec les mathématiques. Il regroupe plus de 2500 scientifiques, majoritairement informaticien · nes ou mathématicien · nes, réparti · es dans plus d'une centaine d'unités en France (laboratoires universitaires, unités mixtes de recherche CNRS, centres de recherches Inria, etc.), voire même à l'étranger.

Les domaines de recherche couverts actuellement par le GdR IFM se structurent en cinq axes thématiques, non disjoints :

Axe 1 : Combinatoire, graphes et systèmes dynamiques

Axe 2 : Calculabilité, complexité, algorithmique et calcul quantique

Axe 3 : Calcul formel, arithmétique et cryptologie

Axe 4 : Programmes, vérification, preuve, automates et logique

Axe 5 : Géométries et image

Les activités du GdR IFM sont structurées en groupes de travail (GT) spécialisés, dont certains communs avec d'autres GdR. La vie scientifique du GdR se déroule au travers des réunions annuelles de ses GT, de ses journées nationales annuelles, de son école dédiée aux jeunes chercheurs et chercheuses, et de diverses actions spécifiques dont des journées transverses, les années thématiques, etc.

La direction du GdR est assurée par ses responsables s'appuyant d'une part sur un comité de direction (ComDir) composé des responsables du GdR, de la présidence du Conseil Scientifique du GdR, et de trois membres nommés, et d'autre part sur le comité exécutif (ComEx) du GdR réunissant l'ensemble des responsables de GT. Le conseil scientifique (CS) est entre autre en charge d'aider la direction du GdR dans la détermination de sa politique scientifique.

Les directions actuelles et passées du GdR s'associent pour profiter de l'anniversaire des 20 ans du GdR afin de remercier vivement l'ensemble des collègues qui, depuis 20 ans, se sont investis dans la vie du GdR. Sans leur investissement au service de notre communauté, rien n'aurait été possible !

Pierre Fraignaud et Guillaume Theyssier

Les vingt ans du GdR IFM, vus du Conseil Scientifique

Algorithmes, logique informatique, complexité, graphes, aléa... les grandes questions et les objets au cœur du GdR IFM sont centraux dans les sciences informatiques. Motivée par des problèmes issus de divers domaines de l'informatique ou parfois d'autres disciplines, leur étude fait souvent appel à des outils et méthodes mathématiques spécifiques, qui demandent des développements propres et différents des approches mathématiques classiques. C'est cette combinaison d'un ancrage au cœur de l'informatique et d'une méthodologie mathématique rigoureuse procurant des garanties qui fédère les communautés scientifiques du GdR IFM.

Des sujets de recherche qui ont diffusé bien au-delà de l'informatique fondamentale.

Les sujets abordés au sein du GdR qui, il y a vingt ans, étaient principalement définis comme des points d'articulation entre informatique et mathématique, irriguent désormais pratiquement toutes les disciplines des sciences informatiques : calcul formel en robotique et en optimisation, géométrie et topologie algorithmique en sciences des données, par exemple.

Les frontières entre le GdR IFM et le reste des sciences informatiques se sont déplacées, et les sujets d'intérêt à l'interface avec d'autres communautés (combinatoire des mots et séquençage génétique, calcul quantique, vérification de programmes, imagerie et informatique graphique...) se sont multipliés, comme en témoigne le rattachement d'une partie des GTs du GdR IFM à un deuxième GdR, souvent plus tourné vers un domaine applicatif [1], ainsi que la participation des chercheurs à plusieurs GdR. Au cours des vingt dernières années, l'informatique fondamentale a réalisé des progrès algorithmiques et en complexité marquants, mais elle a aussi trouvé un retentissement important dans l'ensemble de la société à travers, par exemple, la popularisation du mot « algorithme » et la prise de conscience de l'impact des algorithmes dans la « vie quotidienne » et donc de l'importante problématique de leur transparence, de leur qualité et de leur équité.

Une singularité des liens entre le GdR IFM et les autres domaines de l'informatique, ou d'autres disciplines au premier rang desquelles on trouve les mathématiques, mais aussi la biologie ou la physique, est qu'ils ne se réduisent pas à l'apport de motivations nouvelles, mais sont caractérisés par de multiples « allers-retours ». Par exemple, le lien entre théorie des types et homotopie, introduit il y a vingt ans, a stimulé les interactions entre ces thématiques. Ces travaux ont à la fois constitué un fondement logique essentiel pour le développement d'assistants de preuve et ont aussi attiré des mathématiciens du domaine de la topologie algébrique vers la formalisation de leurs preuves avec un regard renouvelé sur celles-ci. Dans un registre différent, le langage Kappa, langage basé sur des règles de réécriture de graphes avec une syntaxe inspirée de la chimie, fut initialement introduit pour décrire des interactions entre protéines, mais il a depuis permis de modéliser et d'étudier bien d'autres phénomènes,

1. Les GTs Bioss (Biologie systémique symbolique) et Seq-BIM (Séquences en Bioinformatique, Informatique et Mathématiques), sont naturellement également affiliés au GdR BIMM – *Bioinformatique Moléculaire : Modélisation et Méthodologie*, le GT Bioss dépendant aussi du GdR RADIA – *Raisonnement, Apprentissage, et Décision en Intelligence Artificielle*. Les GTs GDMM (Géométrie discrète et morphologie mathématique) et MG (Modélisation géométrique) sont aussi rattachés au GdR IG-RV – *Informatique géométrique et graphique, réalité virtuelle et virtualisation*. Le GT C2 (Codage et Cryptographie) dépend également du GdR Sécurité, alors que le GT Arith (Arithmétique des ordinateurs) constitue l'un des groupes du tout récent GdR C4P – *Calcul : Paradigmes, parallélisme, performance, précision*. Le GT IQ (Information quantique), lui, est aussi impliqué dans le GdR *Technologies Quantiques* et le GdR C4P.

en dynamique des populations, en épidémiologie, et a conduit à la formulation de nouvelles hypothèses biologiques, comme l'identification de bio-marqueurs de la fibrose hépatique. Les liens étroits entretenus entre la combinatoire et la physique statistique, ou les systèmes dynamiques et la physique mathématique en sont également une preuve.

Des interactions fortes entre GTs.

Les GTs étant définis autour de méthodes et d'objets fondamentaux en informatique, leur dénomination a relativement peu varié au cours de ces vingt ans. Mais cette stabilité masque des évolutions importantes. On peut mentionner le développement considérable de certaines communautés, visible à partir de l'effectif des GTs correspondants. C'est le cas notamment du GT Informatique quantique et du GT Codage et Cryptographie. Ce développement est d'ailleurs révélateur de nouvelles interactions entre communautés (et entre GTs), symboles d'un enrichissement mutuel entre domaines au fil des années. Ainsi, la résolution de systèmes polynomiaux, sujet de prédilection du GT Calcul Formel, est devenu un outil de cryptanalyse, développé et adapté à ce contexte particulier au sein du GT Codage et Cryptographie. Les phénomènes aléatoires discrets, étudiés historiquement au sein du GT Aléa, intéressent désormais les GTs Complexité et Algorithmique, Graphes et Informatique Quantique, à travers les algorithmes probabilistes. Ces liens étroits apparaissent clairement sur le graphe des interactions entre GTs, défini par le nombre de membres en commun entre deux GTs, visualisable sur MyGDR.

L'objectif de la suite de ce texte est de mettre en avant, de manière non exhaustive, certaines évolutions scientifiques marquantes du GdR IFM, et notamment de montrer comment des progrès significatifs sur certains sujets, au sein même ou à la frontière de nos disciplines, ont considérablement influencé les questions de recherche abordées au cours des dernières années.

1 Automatisation du raisonnement, des démonstrations et de certains calculs : nouveaux outils, nouveaux sujets

Les thématiques du GdR IFM au cours des vingt dernières années ont été durablement marquées par le développement de divers outils d'automatisation des preuves, de certains calculs et du raisonnement, qu'il s'agisse des assistants de preuve, des solveurs (SAT, SMT, MILP...) ou de systèmes de calcul formel, mais aussi des LLMs dont l'essor a eu un retentissement bien au-delà des sciences informatiques. Ces techniques d'automatisation sont devenues centrales dans nos disciplines, à la fois comme outils et comme objets d'étude.

1.1 La multiplication des usages des assistants de preuve et solveurs

L'apparition la plus retentissante des assistants de preuve en informatique fondamentale remonte à la preuve du théorème des quatre couleurs. Mais, depuis les travaux de G. Gonthier et ses collaborateurs en 2005, les capacités des assistants de preuve ont considérablement augmenté, comme l'a montré le développement de CompCert, un compilateur C optimisant opérationnel et formellement prouvé. Cette maturité a transformé les assistants de preuve en outils essentiels en informatique fondamentale. Ainsi, le besoin de garanties sur l'exactitude et la précision des calculs a naturellement conduit des domaines comme le calcul formel et l'arithmétique des ordinateurs à fréquemment utiliser (et faire progresser) les assistants de preuve, et à développer des liens étroits avec les communautés de preuve formelle.

Les assistants de preuve sont aussi un élément important direct ou indirect de la recherche sur la preuve de programmes, sujet qui va au-delà du périmètre du GdR IFM. Il peut s'agir du développement de bibliothèques dédiées, comme le projet Iris pour la preuve de programmes Rust, ou bien d'un usage en back-end d'outils de vérification déductive de programmes, comme pour le système Why3, ou encore pour renforcer l'assurance de correction de résultats d'articles de recherche en théorie de la programmation. Un autre signe de maturité est l'arrivée sur la scène d'un nouvel assistant de preuve Lean « petit cousin » de Rocq qui, tout en s'appuyant sur des principes analogues, a su attirer des communautés nouvelles en particulier chez les mathématiciens.

Plus généralement, le traitement automatique de certaines parties des preuves par des outils « en boîte noire » s'est généralisé dans de nombreux domaines. C'est le cas par exemple de la méthode de déchargement, étape classique dans diverses preuves de théorie des graphes traitée de manière automatique par le biais de la programmation linéaire. On assiste à un mouvement similaire en cryptographie où l'absence et l'optimisation d'attaques différentielles ou linéaires en cryptographie symétrique est analysée grâce à des solveurs MILP (Mixed-Integer Linear Programming), celles d'attaques dites « algébriques » par des outils de résolution de systèmes polynomiaux issus du calcul formel. Dans toutes ces situations, la phase délicate est devenue la recherche d'une modélisation appropriée du problème considéré qui permette un traitement automatique efficace. Le domaine de l'implémentation, matérielle ou logicielle, des protocoles cryptographiques, repose aussi en grande partie sur des outils automatiques dédiés, tels EasyCrypt, CryptoVerif ou Tamarin.

Plus récemment, l'utilisation de LLM comme aide dans la résolution de certains problèmes de combinatoire, tels certains des célèbres problèmes de Erdös, souligne l'apport de l'IA générative pour identifier la littérature pertinente de manière efficace, ou pour aider à localiser des contre-exemples. Un sujet de recherche très actuel est donc l'utilisation des techniques d'apprentissage automatique dans les assistants de preuve pour guider la stratégie utilisée.

1.2 L'apprentissage, source de nouvelles problématiques en informatique fondamentale

L'apprentissage profond est également devenu un outil essentiel dans divers domaines de l'informatique fondamentale, notamment dans des tâches de modélisation. Il offre par exemple une méthode extrêmement efficace en remplacement de modèles statistiques explicites pour réaliser les attaques dites par canaux auxiliaires, qui exploitent l'analyse de traces d'exécution physique (par exemple la consommation de courant ou le rayonnement électromagnétique émis) pour retrouver des quantités secrètes. Une évolution similaire peut être observée dans les travaux de modélisation de systèmes biologiques, avec l'utilisation de modèles dits substituts (surrogate modeling) pour gagner en efficacité et en capacité de prédiction.

Les techniques d'apprentissage fournissent donc des outils précieux dans certains domaines du GdR. Mais elles sont également devenues des objets d'étude, conduisant la communauté d'informatique théorique à s'intéresser aux algorithmes sous-jacents sous l'angle de la complexité, de la fiabilité ou des modèles de calcul, mais aussi de la confidentialité et de la sobriété.

L'informatique fondamentale a pu apporter des réponses à ces questions dans le cas des heuristiques d'optimisation randomisées (algorithmes évolutionnaires, recuit simulé...), sujet sur lequel les outils théoriques développés depuis la fin des années 1990 comme l'analyse de drift, les techniques d'analyse lissée, ont amené une quinzaine d'années plus tard des résultats-clefs sur la complexité de ces algorithmes et ont influencé directement leur conception. Dans le cas de l'apprentissage profond, les outils de la théorie PAC (Probably Approximately Correct),

les notions de complexité de communication, la théorie de l'information apparaissent comme des éléments essentiels dans les nombreux travaux visant à une meilleure compréhension des performances de ces techniques. Ainsi, la question des limites de l'expressivité des réseaux de neurones profonds peut être abordée en revisitant les résultats classiques de théorie de la complexité des circuits puisque les réseaux de neurones peuvent être vus comme une classe particulière de circuits opérant, non sur des entrées booléennes mais sur des réels.

Garantir certaines propriétés, fonctionnelles ou géométriques, en apprentissage profond est aussi un axe de recherche développé au sein du GdR IFM. De nouveaux composants non-linéaires fondés sur des algèbres $(\max, +)$ sont proposés pour remplacer les circuits convolutionnels, afin d'intégrer du traitement d'image non-linéaire. D'autres composants intègrent les opérateurs d'algèbres géométriques, pour rendre les réseaux invariants sous certaines transformations géométriques. De nouveaux algorithmes d'optimisation permettent enfin d'imposer une fonction implicite lipschitzienne en sortie, et rendent donc les traitements géométriques beaucoup plus stables et prédictibles.

Pouvoir expliquer les résultats produits par certains systèmes d'IA est une autre question essentielle pour laquelle les modèles formels comme la logique ou les automates fournissent des outils pertinents. La théorie algorithmique des jeux est également un domaine fondamental pour comprendre l'IA multi-agent.

La sécurité de l'IA est un autre sujet extrêmement vaste, abordé depuis quelques années par la communauté d'informatique fondamentale, couvrant des questions allant de la certification des logiciels d'IA à la protection de la confidentialité des données manipulées via la cryptographie homomorphe — dont la possibilité, démontrée pour la première fois en 2009, a ouvert de nombreuses perspectives et est désormais explorée par un tissu industriel dense dans lequel le GdR est impliqué. Les besoins en apprentissage automatique, et en HPC, et les évolutions matérielles des fabricants de puces ont également des répercussions dans le domaine de l'arithmétique des ordinateurs. L'utilisation de calculs sur des nombres de 16 bits ou moins soulève des questions liées à l'algorithmique à très petite précision et à la maîtrise de cette précision.

2 Le GdR IFM, moteur dans la mise à disposition de nouveaux logiciels

Les membres du GdR IFM ont joué au cours de ces vingt ans un rôle très important dans la conception et la mise à disposition de la communauté scientifique de nombreux logiciels et librairies spécialisées, devenus d'usage courant dans diverses disciplines. Ces logiciels résultent généralement d'un effort collaboratif soutenu, qui bénéficie à de multiples utilisateurs, au sein de la communauté informatique fondamentale et à l'extérieur. Nombre d'entre eux ont d'ailleurs été récompensés par des prix prestigieux [2]. Cette capacité à développer des logiciels de grande ampleur sur le long terme est une spécificité et une grande force mondialement reconnue de la communauté française d'informatique fondamentale.

On pense bien sûr au logiciel Rocq (anciennement Coq) dont le développement a débuté il y a 40 ans. Comme déjà évoqué, ces vingt dernières années ont été marquées par l'appropriation de cet outil par plusieurs communautés qui l'utilisent pour aller plus loin dans leur discipline,

2. ACM Software System award pour Coq/Rocq, Prix Science Ouverte du logiciel libre de recherche du MESR pour MPFR et Pari/GP, également ACM SIGSAM Richard Dimick Jenks Memorial Prize pour Pari/GP, Prix Levchin pour CADO-NFS, Test of Time award du Symposium on Computational Geometry pour CGAL, Software award du Symposium on Geometry Processing pour DGtal...

par des travaux théoriques pour faire évoluer le langage autour de la théorie homotopique des types qui a permis de mieux comprendre la structure des preuves d'égalité en les interprétant comme des chemins transformant un objet en un autre qui lui est égal et qui a de nombreuses répercussions en terme théorique et en terme d'outils.

Il est impossible de citer toutes les autres réalisations logicielles notables impliquant des membres du GdR. Parmi ces travaux, on peut notamment mentionner l'avènement de SageMath, un système de calcul mathématique open source et collaboratif, qui offre une interface de programmation simple en Python et facilite l'accès à de très nombreux composants logiciels autour du calcul numérique et symbolique, comme le système Pari/GP ou la bibliothèque spécialisée efficace MPFR. Des membres du GT Calcul Formel ont activement participé au développement du système et de ses composants (FLINT, fplll, FFLAS-FFPACK, msolve, etc.), ainsi qu'à sa diffusion dans le monde de la recherche et de l'enseignement.

Le logiciel CADO-NFS, seule implémentation disponible au monde de l'intégralité de l'algorithme du crible algébrique pour le calcul du logarithme discret et la factorisation, a permis de battre les records (toujours en vigueur) de taille des entiers pour lesquels ces deux problèmes ont pu être résolus. Mais il est aussi utilisé en boîte noire dans des attaques sur des protocoles cryptographiques ou dans des travaux mathématiques.

En géométrie, la bibliothèque CGAL, dont le développement a été initié à la fin des années 1990, est une contribution majeure de la communauté française de géométrie algorithmique, désormais référence mondiale du domaine, qui ne cesse d'évoluer et possède plus de 200 utilisateurs commerciaux dans le monde, et compte plus de 10 000 téléchargements par an. La bibliothèque DGtal, initiée par la communauté française de géométrie discrète, s'est, elle, imposée comme la bibliothèque de référence dans la communauté académique internationale depuis 10 ans. Ces bibliothèques d'informatique géométrique ont accompagné l'essor du traitement géométrique des données numériques, avec des applications en CFAO, impression 3D, architecture, maillage pour le calcul scientifique, analyse d'images bio-médicales et matériaux. Les recherches autour de l'analyse topologique des données, sujet introduit il y a une vingtaine d'années, ont fourni un nouveau cadre robuste, alternatif aux algorithmes d'apprentissage pour analyser les données. Ces travaux ont également donné lieu à des applications dans de multiples domaines (visualisation des données, sciences des matériaux, et même à l'optimisation du recrutement des joueurs en NBA [3]). Ils se sont notamment matérialisés à travers le logiciel Gudhi.

3 L'informatique fondamentale à l'épreuve du réel ?

Analyser les performances des algorithmes indépendamment de propriétés spécifiques vérifiées par les données auxquelles ils s'appliquent est un objectif historique de l'informatique théorique. Cette approche fournit des garanties universelles, atteste du caractère généraliste d'un algorithme et dispense ses utilisateurs d'avoir à identifier les types d'entrées qui sont pertinents pour leurs applications. Toutefois, cette analyse dans le pire cas ne reflète pas toujours le comportement de ces algorithmes et ces systèmes dans la « vraie vie », car elle fait parfois la part belle à des exemples pathologiques qui ne sont pas pertinents en pratique.

La référence illustrant le mieux cette situation est l'algorithme du simplexe en programmation linéaire dont la complexité dans le pire cas est exponentielle en le nombre de variables, alors qu'en pratique, la croissance de son temps de calcul reste modeste, au point qu'il est plus efficace que la méthode de l'ellipsoïde qui, elle, est pourtant polynomiale dans le pire

3. Cf. article du New-York Times de mars 2012.

cas. Ce fossé entre le pire cas et la pratique a été comblé par Spielman et Teng grâce à l'introduction d'un nouveau cadre d'analyse, l'analyse lissée, montrant que si l'on perturbe une instance quelconque (même pathologique) par un petit bruit aléatoire gaussien, alors on obtient un temps d'exécution polynomial. Autrement dit, les instances du pire cas sont fragiles, et les données du monde réel, qui contiennent toujours de légères imprécisions ou perturbations, conduisent à de bonnes performances.

Ce type d'analyse, introduit pour la première fois il y a 25 ans, a ouvert une perspective nouvelle et fructueuse en informatique théorique. L'idée générale consiste à identifier des propriétés des données du monde réel et à les exploiter pour apporter des garanties rigoureuses sur l'algorithme quand il opère sur des entrées ayant ces propriétés particulières. L'identification des structures pertinentes (parcimonie, symétrie..) a parfois lieu en interaction avec d'autres disciplines, par exemple la robotique ou la cryptographie dans le cas de la résolution de systèmes polynomiaux. Diverses techniques d'analyse de complexité, qui reflètent mieux la pratique, ont donc fleuri au cours des vingt dernières années : la complexité lissée mentionnée précédemment, la complexité paramétrée qui différencie les instances en fonction d'un ou plusieurs paramètres caractéristiques, typiquement la largeur arborescente d'un graphe ou la twin-width, quantité récemment identifiée par des membres du GT Graphes.

Comparer les performances des algorithmes rapides est également devenu un sujet de recherche important, ceux-ci étant désormais confrontés à des données de très grande taille. Dans ce but, la complexité à grain fin fournit une information plus précise sur la complexité des problèmes polynomiaux en reliant leur difficulté de manière à préserver le degré des polynômes en jeu. Ce cadre permet donc de distinguer plusieurs grandes catégories au sein des problèmes polynomiaux. La complexité à grain fin fournit également des bornes inférieures précieuses sur la complexité d'un grand nombre de problèmes (calcul du diamètre d'un graphe pondéré ou non, du nombre chromatique...) en se ramenant à un petit nombre d'hypothèses communément admises. De tels résultats démontrent ainsi l'optimalité de certains algorithmes.

L'essor du Big Data a conduit à l'analyse de données de plus en plus massives et disparates. L'approche standard est de les placer en très grande dimension, puis de leur trouver une cohérence interne en les approchant par des variétés de dimension intermédiaire. Afin de mieux comprendre leurs propriétés, de nombreuses techniques d'analyse topologique ou géométrique des données efficaces ont dû être développées : réduction de dimension, construction de complexes « nerf », analyse robuste de la topologie. Ces travaux sont aussi utilisés pour caractériser les espaces latents de certains réseaux de neurones profonds, afin d'expliquer le rôle des paramètres essentiels.

Dans ce même souci de mieux prendre en compte les contraintes du monde réel dans la conception et l'analyse des algorithmes, divers modèles de calcul plus pertinents dans certains contextes ont été définis, par exemple quand les données arrivent de façon séquentielle et doivent être traitées à la volée, sans pouvoir être stockées. Les modèles distribués ont également été sources de nombreuses avancées dans le domaine de la complexité et de l'algorithmique.

La prise en compte de nouvelles structures de données, mieux adaptées au monde réel, a aussi fortement marqué la théorie des bases de données : elle a été amenée à définir et à manipuler de nouveaux modèles de données sous forme d'arbres (comme dans XML) ou de graphes (comme dans RDF), suscitant alors de nouveaux travaux en algorithmique, en complexité, mais aussi sur les langages de requête.

4 De la fiction aux perspectives concrètes du calcul quantique

Fiction introduite au début des années 1980, le calcul quantique a initialement éveillé l'intérêt grâce aux algorithmes de Shor et de Grover à la fin du siècle dernier, puis plus récemment avec l'algorithme de Harrow-Hassidim-Lloyd et la méthode QSVT [4] pour la résolution de systèmes linéaires. Suite à la conjonction de ces progrès théoriques et de nouvelles perspectives technologiques, un énorme effort financier public et privé a été consenti sur ce sujet dans de nombreux pays depuis une dizaine d'années, incarné en France par la stratégie nationale quantique [5], ce qui a considérablement développé et transformé la communauté scientifique, académique et industrielle. L'émergence de programmes de R&D sur le sujet dans de nombreuses grandes entreprises et l'apparition de multiples startups ont été rendues possibles grâce au rôle moteur d'un grand nombre de docteurs mais aussi de chercheurs confirmés issus de la recherche publique. La croissance fulgurante de la communauté internationale peut se mesurer par le nombre de soumissions à la conférence annuelle QIP, passé de 160 en 2006 à 700 vingt ans plus tard.

Les perspectives ouvertes par le calcul quantique, attisées à la fois par la promesse ultime d'un ordinateur quantique et par l'arrivée de prototypes très bruités (NISQ), touchent un très grand nombre de domaines de l'informatique fondamentale : nouveaux modèles de calcul, nouvelle hiérarchie de complexité, nouveaux algorithmes, nouvelles attaques en cryptographie, nouveaux langages de programmation... Certaines de ces questions sont au cœur du GT Informatique Quantique, d'autres sont à l'interface avec d'autres GTs.

De nombreux sujets nouveaux sont naturellement apparus en complexité et algorithmique, incluant par exemple la classification des problèmes selon leur complexité en requêtes ou encore de communication. Des résultats de séparation ont ainsi permis de clarifier les frontières entre calcul classique et calcul quantique, offrant des bases solides pour identifier des tâches pour lesquelles on pourrait apporter une preuve rigoureuse que les algorithmes quantiques offrent un avantage. Mais ce sont aussi de nouvelles méthodes qui ont été développées et qui, peu à peu, ont influencé des domaines non quantiques : en algorithmique (avec la déquantisation d'algorithmes quantiques ou la mise en évidence de l'absence de solutions par programmation linéaire pour le problème du voyageur de commerce), en théorie des codes correcteurs d'erreurs (par exemple les codes localement décodables), ainsi qu'en complexité classique, en particulier la réduction quantique de Regev qui relie la difficulté moyenne de Learning With Errors (LWE) à la difficulté dans le pire des cas de problèmes sur les réseaux euclidiens.

Des algorithmes quantiques sont proposés dans de nombreux domaines, pour la simulation de systèmes quantiques, pour la chimie quantique... mais aussi en cryptanalyse pour attaquer d'autres problèmes que ceux résolus par l'algorithme de Shor, par exemple des constructions classiques de codes d'authentification de messages. La recherche d'algorithmes quantiques en cryptanalyse est tout particulièrement importante pour évaluer la sécurité des systèmes *post-quantiques* en cours de standardisation. Censés remplacer les systèmes à clef publique classiques reposant sur la difficulté du logarithme discret et de la factorisation, ces alternatives n'ont évidemment d'intérêt que si on acquiert la conviction qu'elles résistent à un adversaire quantique. Les technologies quantiques ouvrent également des nouvelles possibilités en cryptographie en permettant de réaliser des fonctionnalités inatteignables classiquement, grâce à l'impossibilité de cloner l'information. Auparavant limitées à la distribution de clefs

4. Quantum Singular Value Transformation.
5. `https://quantique.france2030.gouv.fr/`

inconditionnellement sûre, les fonctionnalités visées se sont considérablement enrichies au cours des dernières années, apportant par exemple la possibilité d'évaluer un programme tout en empêchant sa copie.

Parmi les briques essentielles à la construction d'un ordinateur quantique, la correction d'erreurs joue un rôle important car l'information quantique est rapidement corrompue par le bruit. Il faut donc corriger les erreurs plus vite qu'elles se créent. La recherche de bons codes quantiques permettant de réaliser des circuits quantiques tolérants aux fautes avec un surcoût constant a suscité de nombreux travaux au cours des dix dernières années, et a également introduit des concepts qui se sont avérés centraux en complexité quantique.

L'impossibilité de cloner l'information quantique rend l'écriture même de programmes quantiques délicate, et complique également leur vérification par des techniques classiques de débuggage. La recherche de langages de programmation adaptés, et d'outils de vérification suscite donc depuis quelques années de nombreux travaux, en particulier sur la sémantique des langages de programmation quantiques et la représentation des circuits quantiques, notamment par des formulations du calcul quantique dérivées de la théorie des catégories.

5 L'omniprésence des probabilités

Les probabilités sont par essence au cœur de constructions d'objets en combinatoire, de l'analyse d'algorithmes en moyenne, des méthodes de recuit simulé, d'apprentissage, et bien entendu du calcul quantique. Mais elles sont aussi, au cours de ces vingt ans, devenues centrales dans la plupart des autres domaines de l'informatique fondamentale, au point que le GdR IFM a organisé une année thématique sur le sujet en 2023-24 [6]. Ainsi, toutes les thématiques où la notion d'aléa est essentielle manipulent des probabilités, comme la calculabilité, les algorithmes randomisés, les systèmes dynamiques... C'est également le cas de la correction d'erreurs et de la théorie de l'information qui modélise du bruit affectant les données.

Plus généralement, la gestion de l'incertitude est devenue centrale pour tous les GTs du GdR IFM. Il s'agit de quantifier l'incertitude sur les données, du fait par exemple de données incomplètes ou bruitées, situations classiques en bases de données, ou pour l'inférence géométrique, mais aussi l'incertitude intrinsèque à beaucoup de systèmes distribués ou cyber-physiques étudiés en vérification. L'étude de ces systèmes a nécessité le développement de modèles probabilistes, et a suscité nombre d'analyses pour évaluer la robustesse des algorithmes : comprendre ce qui demeure calculable malgré les perturbations, apporter des garanties sur le résultat dans un contexte bruité. Cette même approche probabiliste apparaît naturellement dans l'étude des algorithmes randomisés, mais aussi dans le contexte de l'analyse lissée qui perturbe artificiellement les instances pour analyser le comportement d'un algorithme hors de certains cas pathologiques. La théorie des probabilités est également devenue un outil essentiel dans la démonstration de résultats importants, comme celle de la conjecture d'Erdös-Faber-Lovász [7], cinquante ans après sa formulation.

La théorie des probabilités a également fait son apparition en vérification de programmes, de systèmes informatiques, de multiples façons. On y vérifie des systèmes informatiques ayant un comportement probabiliste — chaînes de Markov, processus de Markov partiellement observables, jeux stochastiques — et dans ce cadre, au-delà de la question de savoir si une

6. https://gdr-ifm.fr/thematic-years/probabilities
7. Selon laquelle tout graphe correspondant à l'union de n cliques de taille n dont l'intersection deux-à-deux contient au plus un sommet, a un nombre chromatique inférieur ou égal à n.

propriété donnée d'un système est vérifiée ou non, on calcule ou on estime la probabilité que cette propriété soit vraie. Les méthodes de vérification statistiques sont apparues, dans lesquelles on échantillonne des traces d'exécution pour en déduire des garanties approchées, rapidement. La recherche en sémantique des langages de programmation s'est développée, elle, d'une part pour traiter des langages de programmation statistiques, dont la création répond aux besoins de description de distributions et d'algorithmes d'échantillonnage complexes, et d'autre part pour établir les bases de futurs langages de programmation quantique.

6 Quand discret et continu se rejoignent

Historiquement, l'informatique fondamentale s'est construite autour d'objets discrets (mots, graphes, automates, circuits...) et de modèles finis. Les outils fournissant des garanties d'exactitude sont donc longtemps restés de nature discrète, même si l'importance croissante de la théorie des probabilités a fait entrer des outils de nature continue dans plusieurs disciplines, via des modèles probabilistes, des approximations ou des métriques continues, comme l'entropie en théorie de l'information.

Mais cette interaction avec le continu est devenue de plus en plus riche au cours des vingt dernières années, à travers de nouveaux outils et de nouveaux objets d'étude. Ainsi de nouveaux modèles de calcul manipulant des quantités continues, parfois mêlées avec du discret, ont vu le jour, visant à explorer ce qu'il est possible de calculer en utilisant l'information quantique, des composants analogiques, des modèles biologiques... Le continu est par exemple au cœur du calcul quantique, qui manipule des états dans des espaces de Hilbert où la notion d'erreur est elle-même continue et non plus discrète comme en classique. Il est aussi au cœur des réseaux neuronaux, qui calculent et optimisent certaines fonctions sur $\mathbb{R}^n$, pour de très grandes valeurs de n.

De plus, l'essor de l'apprentissage statistique et des algorithmes opérant sur les réels a conduit à formuler pour le calcul continu des questions de pouvoir de calcul et de complexité identiques à celles du cadre discret classique. C'est ainsi que des travaux récents ont par exemple montré comment on pouvait simuler des calculs sur des réels par des équations différentielles polynomiales, de manière à pouvoir exprimer des ressources classiques, telles que le temps de calcul, par des propriétés de ces équations, comme la longueur de la courbe.

Ce glissement vers le continu apparaît également en cryptographie, où les systèmes à clef publique reposant sur l'arithmétique modulaire sont peu à peu supplantés par des systèmes reposant sur les réseaux euclidiens, en cryptographie post-quantique ou pour le chiffrement homomorphe : si les objets manipulés restent discrets, c'est leur plongement dans $\mathbb{R}^n$ et donc des propriétés géométriques dans un espace continu de grande dimension qui sont à l'origine de la difficulté des problèmes sous-jacents.

Le problème de représentation d'un objet continu par un objet discret est au cœur des problèmes étudiés en géométrie informatique. L'approximation convergente des données, souvent simple échantillonnage d'objets réels continus, se fait assez naturellement par des primitives finies (mailles, surfaces splines). Ces dernières années ont vu l'émergence de nombreuses techniques pour définir une géométrie différentielle convergente sur ces objets finis. De façon plus profonde encore, la théorie géométrique de la mesure (cycle normal, courants normaux corrigés), adaptée aux données réelles, offre maintenant un cadre unifié pour les représentations finies et continues, garantissant stabilité des mesures et calcul efficace.

La combinatoire est riche d'exemples de ce mouvement d'aller-retour entre discret et continu. Lorsque la taille d'un système aléatoire atteint l'infini, un objet discret peut se transformer en un objet continu. Les phénomènes limites continus sont souvent plus parlants

que les phénomènes discrets et apportent des réponses universelles ; le mouvement brownien en est l'illustration, et ce en particulier dans l'étude des cartes planaires (ce sont des graphes dessinés sur une surface orientée). Celles-ci sont en particulier source de modèles pour la gravitation quantique, dont l'objet est de définir un espace-temps relativiste et quantique : les limites de cartes de grande taille décrivent la théorie quantique de Liouville.

Les langages de programmation étudiés classiquement en sémantique, et même les langages de programmation probabilistes, ne considéraient traditionnellement que des distributions discrètes. Au vu des besoins croissants de formalismes de description de distributions statistiques complexes, un glissement s'est opéré vers le continu, avec l'émergence de langages de programmation dits statistiques : on peut y tirer des objets au hasard dans des domaines continus, typiquement $\mathbb{R}^n$, mais aussi dans des espaces de fonctions d'ordre supérieur ou de distributions, qui seront donc elles-mêmes aléatoires ; un exemple typique en est donné par les processus de Dirichlet.

L'hybridation entre approche numérique et approche symbolique est aussi devenu un axe important, à la croisée des problématiques d'optimisation, d'approximation et d'algorithmique numérique. Elle repose sur l'idée d'exploiter la rapidité des méthodes numériques pour accélérer certains calculs, tout en préservant la capacité à manipuler des objets symboliques de manière exacte. Cette démarche s'accompagne de techniques de certification, qui garantissent la validité mathématique des résultats obtenus. Cette stratégie se révèle efficace lorsqu'elle est combinée à la méthode homotopique pour le calcul d'approximations de racines d'équations polynomiales. Elle trouve également des applications dans l'étude et la manipulation d'opérateurs différentiels. Elle est mobilisée pour le calcul d'intégrales intervenant dans l'estimation des probabilités de collision entre satellites et débris en orbite. Un algorithme fondé sur ces principes a notamment été embarqué sur un mini-satellite expérimental de l'Agence spatiale européenne.

Conclusion

Au-delà de ces quelques évolutions marquantes, le recueil des documents rédigés par chaque GT donne une vision kaléidoscopique de ces deux décennies de recherche en informatique théorique, vision qui témoigne à la fois de la richesse et du foisonnement de sujets abordés, et d'une approche commune où aspects fondamentaux, implémentation et mise en œuvre dans de nombreux domaines sont toujours étroitement mêlés. En particulier, il apparaît clairement que les concepts, les formalismes et les outils de l'informatique fondamentale jouent un rôle structurant dans l'ensemble des sciences informatiques, et plus généralement dans tous les domaines recherchant une analyse rigoureuse et des garanties sur l'optimalité, la qualité des résultats d'un algorithme et sur sa mise en œuvre. Il est donc essentiel que la mise en avant de quelques sujets (IA, informatique quantique, cryptographie) qui bénéficient actuellement d'une forte couverture médiatique et de financements dédiés conséquents ne se fasse pas au détriment d'autres aspects, et qu'elle ne compromette pas l'équilibre entre fondements théoriques et avancées technologiques, entre recherche pilotée et recherche guidée par la curiosité, dont l'alliance est à l'origine des succès scientifiques et technologiques de ces vingt dernières années.

Les membres du Conseil Scientifique du GdR IFM.

Valérie Berthé, Anne Canteaut, Jérémie Chalopin, Jean Goubault-Larrecq, Emmanuel Jeandel, Jacques-Olivier Lachaud, Frédéric Magniez, Anca Muscholl, Christine Paulin-Mohring, Sophie Tison, Ioan Todinca, Gilles Villard.