

Nathan Tournoux

INGÉNIEUR LOGICIEL HPC - DOCTEUR EN PHYSIQUE SUBATOMIQUE

INTRODUCTION

Ingénieur logiciel HPC et docteur en physique des particules avec six ans d'expérience, je conçois et développe des logiciels hautes performances au format ligne de commande ou application native/WEB/mobile. J'ai participé à plusieurs projets de recherche en laboratoire et de R&D en startup, où la rigueur, la fiabilité et les performances sont essentielles. Je maîtrise l'architecture logicielle, le calcul parallèle et les communications chiffrées TLS. Habitué à travailler en autonomie ainsi qu'en collaboration internationale.

COMPÉTENCES TECHNIQUES

Langage	Rust, Haskell, Python, Java, C++, Lua, Bash, GLSL, WGSL
Framework	WPGU, egui, burn, OpenGL, OpenCL, SDL2
Outils	Git, Helix, Vim
Sécurité	SSL
Système d'exploitation	MacOS, Linux (ArchLinux, Debian, Ubuntu), Windows
Méthode	Scrum, KANBAN
Autre	Programmation parallèle (CPU et GPU)

COMPÉTENCES FONCTIONNELLES

Communication scientifique	Capacité à présenter un sujet et des résultats de façon concise (<10 min) ou exhaustive (>1h) devant un public expert (conférences internationales) ou débutant (festival, cours)
-----------------------------------	---

EXPÉRIENCE PROFESSIONNELLES

2024 – 2025 **Concepteur développeur Fullstack**

LETSCAN

 NANTES, FRANCE

Dans la section R & D dédiée au traitement efficace du signal, j'ai contribué à la création d'une application dans le but d'une création de startup.

- **Équipe:** Leader technique dans une équipe de trois développeurs. Travail à distance avec une personne à Nantes et deux personnes à Ancy.
- **Méthodologie:** Méthode de gestion projet Scrum (sprint de 3-4 semaines) avec Kanban
- **Contexte:** LetScan s'est focalisé sur la maintenance de câbles par analyse de signaux inductifs dans des câbles métalliques. Après de nombreuses années concentré sur un seul sujet, LetScan souhaite étendre ses horizons en développant une méthode d'analyse et des outils plus performants et à usages plus variés.
- **Objectif:** Créer des applications natives et mobiles de démonstration des capacités et usages de la méthode d'analyse développée par LetScan, puis présenter ces applications à des investisseurs, en particulier lors de la conférence AI Everything Global 2025.
- **Tâches réalisées:**
 - Développement d'un backend CPU et GPU pour le traitement efficace du signal en Rust.
 - Création et implémentation d'un serveur de calcul en Rust, d'un client web et d'une application native avec egui et cpal, connectés par des liaisons chiffrées TLS pour analyser et synthétiser des signaux.
 - Création de deux serveurs Archlinux sur Hostinger et chiffrement des partitions mémoire. Un serveur pour héberger le GitLab et un deuxième pour héberger les serveurs backend de calcul. Installation des certificats SSL pour les connections TLS chiffrées avec le client web.
 - Édition du code source avec l'éditeur modal Helix, et versionnage avec l'outil git et un serveur GitLab interne.
 - Création d'une application mobile en Rust avec egui dédiée à la classification des émotions des bébés. Pour ce faire, un modèle d'IA a été entraîné sur les données issues de la méthode d'analyse de signal développée par l'entreprise, puis intégré à l'application avec le framework Burn.
 - Démarchage et présentation des nouveaux produits développés auprès d'investisseurs lors de la conférence internationale AI Everything Global 2025.
- **Journée type:**
 - Correction des régressions repérées par les tests unitaires ou les tests en utilisation.
 - Conception et développement du backend et du frontend en Rust avec egui.
 - Daily de 15min pour faire un point sur les tâches de la journée et du sprint en cours
- **Résultats obtenus:**
 - Nouveau module de traitement du signal fonctionnel et de l'ordre de 100 fois plus efficace que les prototypes antérieurs de l'entreprise.
 - Production des applications de démonstration dans les délais avant la conférence AI Everything Global 2025, ce qui a débouché sur plusieurs discussions de financements.

Stack technique	Rust (egui, WGPU, cpal, burn, futures, rayon, WASM), SSL/TLS, NextCloud (Deck), Git, GitLab, LUKS
------------------------	---

Dans le contexte de ma thèse sur l'étude du plasma de quarks et de gluons, j'ai conçu une nouvelle méthode de simulation et développé un logiciel haute performance pour mettre cette méthode en pratique et montrer ses avantages.

- **Équipe:** Doctorant dans une équipe de quatre personnes avec trois docteurs, dont deux avec le rôle d'encadrant.
- **Méthodologie:** Méthode de gestion de projet Scrum (sprint de 1-2 semaines)
- **Contexte:** L'étude du plasma de quarks et de gluons (fluide à températures extrêmes) produit dans les accélérateurs de particules est notoirement difficile, en particulier d'un point de vue théorique. Des méthodes de simulation numérique ont donc été développées pour approximer les résultats en première approche. Les plus précises de ces simulations nécessitent en général des mois de simulations dans des centres de calcul. Des méthodes plus précises et efficaces sont donc préférables.
- **Objectif:** Développer une nouvelle méthode de simulation à la fois plus précise et plus efficace que les méthodes existantes pour le cas spécifique du plasma de quarks et de gluons.
- **Tâches réalisées:**
 - Concevoir une nouvelle méthode d'intégration numérique spécialisée pour le contexte du plasma de quarks et de gluons.
 - Optimiser la méthode dans le cas de la simulation de fluides en trois dimensions d'espace.
 - Développer en Rust un logiciel de simulation de fluide incluant la nouvelle méthode.
 - Édition du code source avec l'éditeur modal Helix, et versionnage avec l'outil git et Github.
 - Intégrer ce nouveau logiciel au sein d'un framework incluant toutes les étapes de simulations correspondant aux accélérateurs de particules.
 - Effectuer des simulations complètes sur le centre de calcul du laboratoire YITP en utilisant l'outil d'ordonnancement de tâches Slurm pour soumettre les dizaines de milliers de tâches nécessaires.
 - Analyser les résultats à l'aide de Python (matplotlib), aussi bien la précision des résultats physiques que l'efficacité et la stabilité des algorithmes.
 - Rédiger en LaTeX et publier la nouvelle méthode et les résultats dans un journal.
 - Rédiger en LaTeX et publier une thèse auprès de l'IMT Atlantique et de l'université d'Osaka.
- **Journée type:**
 - Conception et développement de la méthode, ou rédaction du papier/thèse
 - Daily de 5-10min (en général le midi).
- **Résultats obtenus:**
 - Le logiciel de simulation codé en Rust est stable et produit les résultats attendus.
 - La nouvelle méthode produit des résultats aussi précis en la moitié du temps habituel.
 - La précision de cette nouvelle méthode est quadratique par rapport au temps de calcul, ce qui veut dire que doubler le temps de calcul rendra les résultats quatre fois plus précis, là où pour les méthodes existantes les résultats ne seraient que deux fois plus précis.
 - Les résultats et la méthode ont été publiés en 2024 (cf. Publications en dernière page).

Stack technique

Rust (Clap), Python, Helix, Slurm, Git, Github

Dans le cadre de mon stage de master, j'ai créé et implémenté en Rust un solveur d'équations aux dérivées partielles (EDP) sur GPU.

- **Équipe:** Stagiaire avec une encadrante.
- **Méthodologie:** Aucune
- **Contexte:** Le supposé point critique de la matière QCD présente des particularités notoires en termes de fluctuations qui peuvent être étudiées statistiquement. Les types d'équations qui décrivent ces phénomènes sont des équations de diffusion et relaxation stochastiques, qui prennent la forme d'EDP. Une étude purement mathématique s'avère difficile, ce pourquoi les chercheurs s'orientent vers des simulations numériques. Ces simulations impliquent de l'aléatoire, ce qui demande de larges ressources en calcul. En général, ces simulations sont effectuées en une seule dimension d'espace (bien que le monde réel soit en trois dimensions) par limitation en termes de performances.
- **Objectif:** Étudier différentes EDP capables de décrire les phénomènes du point critique de la QCD. Concevoir et développer un logiciel de simulation capable de résoudre des EDP stochastiques (qui font intervenir de l'aléatoire) en trois dimensions d'espace, le tout dans un temps de calcul raisonnable.
- **Tâches réalisées:**
 - Conception d'un format texte pour décrire les équations à résoudre dans le cadre physique (nombre de dimensions, volume du système à résoudre, ...)
 - Implémentation d'un logiciel en Rust capable de lire le format conçu et de produire une représentation intermédiaire d'équations dans le but de produire du code dans le futur.
 - Conception et implémentation d'un compilateur d'EDP pour GPU en Rust. En utilisant la représentation intermédiaire des EDP, du code OpenCL est produit pour à la fois résoudre ces équations et analyser les données produites.
 - Utilisation du logiciel produit pour résoudre différentes EDP et comparer aux résultats mathématiques connus ou aux attentes expérimentales.
 - Utilisation de l'outils d'ordonnancement Slurm pour soumettre des milliers d'exécution du logiciel sur le cluster de calcul GPU de l'in2p3.
 - Rédaction en LaTeX d'un rapport de 25 pages sur le contexte, la méthodologie et les résultats obtenus.
 - **Journée type:** Conception et implémentation en autonomie (période COVID).
- **Résultats obtenus:**
 - Le logiciel supporte de nombreuses EDP en une, deux et trois dimensions.
 - Le compilateur de code GPU fonctionne correctement et produit du code efficace
 - Le calcul parallèle sur GPU grâce au code OpenCL produit a permis d'effectuer des millions de simulations en trois dimensions par semaine, ce qui est nécessaire pour la précision statistique requise dans ce contexte physique.
 - Les résultats produits par ce nouveau logiciel reproduisent correctement les résultats mathématiques connus.
 - Ce logiciel a plus tard été en collaboration avec d'autres chercheurs pour produire des résultats au-delà de ceux mathématiques connus. Ces résultats ont été publiés en 2025 (cf. Publications en dernière page).

Stack technique	Rust (Clap), OpenCL, Helix, Slurm, Git, Github
------------------------	--

PROJETS PERSONNELS

Projets récents

- boxarray (<https://crates.io/crates/boxarray>) : publication sur crates.io de "boxarray", une bibliothèque de code bas niveau en Rust pour allouer des tableaux de taille fixe dans la heap.
- eriun (<https://github.com/xayon40-12/eriun>) : développement d'un langage de preuve basé sur la théorie des types avec parser et type checker en Haskell et coloration syntaxique avec treesitter.
- phase (<https://github.com/xayon40-12/phase>) : simulation de transition de phase au format WEB avec accélération GPU intégralement codée en Rust (dont les shaders) avec egui, WGPU et RustGPU. Une démo WEB est disponible sur Github Pages : <https://xayon40-12.github.io/phase>.

Développement de jeux vidéos

- Rogue-like 3D avec génération procédurale de terrain en Java LWJGL, OpenGL et GLSL.
- Battle Royale 2D en réseau en Java Swing.
- Sokoban 2 joueurs en réseau en C++ SDL2.

Développement de simulations et moteurs

- Boids (nuée d'oiseaux sous forme de nuage de points) en 3D en C++ OpenGL.
- Jeu de la vie avec l'algorithme optimisé HashLife en C++ SDL.
- Moteur graphique par lancé de rayons (ray marching, ray casting) en C++ OpenGL pour simuler de l'optique (réflexion, réfraction) ou pour afficher des fractales 3D.

Compétition de code

- Trois fois vainqueur par équipe du hackathon Cap@Hack de Capgemini à Nantes.
- Niveau 17 et top 2% sur CodinGame (<https://www.codingame.com/profile/22dcb89a9dda3a75ae051973fcf1efa55251933>)
- Participation aux éditions 2017, 2018, 2019 et 2020 de Advent of Code (<https://adventofcode.com>) en Java, Kotlin, Rust et Haskell respectivement.

Stack technique	Rust (WGPU, RustGPU, egui), C++ (SDL/SDL2), Haskell, Java (Swing, LWJGL), Kotlin, OpenGL, Git, Github
------------------------	---

FORMATION

Diplômes

2020 - 2025 **Doctorat en Physique : dynamique des fluides numérique**

IMT ATLANTIQUE, SUBATECH, OSAKA UNIVERSITY, YITP

📍 NANTES, FRANCE ET OSAKA ET KYOTO, JAPON

Doctorat en double diplôme entre la France et le Japon. Boursier MEXT du gouvernement japonais.

2018 - 2020 **Master en Physique des particules**

UNIVERSITÉ DE NANTES

📍 NANTES, FRANCE

Formations

2022 **Sensibilisation / formation à l'intégrité scientifique**

Publications

2025 N. Attieh, **N. Touroux**, M. Bluhm, M. Kitazawa, T. Sami, and M. Nahrgang, "Renormalized critical dynamics and fluctuations in model A in the Hohenberg-Halperin classification", *Phys. Rev. C* 111(2):24906, 2025, doi: 10.1103/PhysRevC.111.024906.

2024 **N. Touroux**, M. Kitazawa, K. Murase, and M. Nahrgang, "Efficient Solver of Relativistic Hydrodynamics with an Implicit Runge-Kutta Method", *PTEP* 2024(6):63, 2024, doi: 10.1093/ptep/ptae058.

Langues

Français Courant



Anglais Courant



Japonais Débutant

