

# Oleksandr Gituliar, PhD

Date of birth: 7 April, 1987

Nationality: Ukrainian

Current city: Copenhagen (Denmark)

Languages: English and Polish (fluent), Russian and Ukrainian (native)

---

## Experience

- 2020.11 – present **Market-Risk Quant (Senior), Danske Bank (Denmark).**
- Productionise Differential Machine Learning for calculating future exposure in C++, AAD, and Python
  - Extend internal Derivatives Scripting Language to learn and predict future prices of derivative instruments in C++
  - Implement regulatory Counterparty Credit-Risk Model in C++
- 2018.07 – 2020.10 **XVA Quant (Associate), Credit Suisse (Poland / UK).**
- Develop exposure models for IR derivatives in C++ / AAD
  - Build F# infrastructure to backtest Initial-Margin Model and report to ISDA and NFA. Develop Excel tools for traders to explain IM
  - Superwise IT on implementing LIBOR transition and oversee books migration
  - Daily support for XVA desk (SQL, Python, common sense)
- 2016.10 – 2018.06 **Postdoc, Hamburg University (Germany).**
- 2015.01 – 2016.09 **Postdoc, Institute of Nuclear Physics (Poland).**
- 2014.04 – 2014.12 **Postdoc, German Electron Synchrotron DESY (Germany).**
- See [my profile](#) on Google Scholar (22 papers, 320+ citations)

---

## Education

- 2009.10 – 2014.06 **PhD in Particle Physics, Institute of Nuclear Physics (Poland),**  
Thesis: *Higher-order corrections in QCD evolution equations and tools for their calculation*,  
[arXiv:1403.6897](#)
- 2004.09 – 2009.06 **MSc in Particle Physics, Dnipropetrovsk National University (Ukraine),**  
Thesis: *Heavy neutral vector boson search in the LHC experiment*

---

## Skills

- Mathematics: **Finite Difference, Monte-Carlo, Linear Algebra, Probability, Differential Equations, Machine Learning**
- Programming: **C++, F#, Python, Git, SQL, Linux**

---

## Software Projects

**Kwinto** — C++/GPU implementation of the Finite-Difference method for pricing derivatives. Sources at [github.com/gituliar/kwinto-cuda](https://github.com/gituliar/kwinto-cuda).

**Fuchsia** — Python implementation of the  $\epsilon$ -form method for solving Differential Equations in symbolic form. Sources at [github.com/gituliar/fuchsia](https://github.com/gituliar/fuchsia).  
Published with [150+ citations](#) in [Comput.Phys.Commun.](#) 219 (2017) 329-338