

QuantLib Python 实作班

使用 C++ DLL/GPU CUDA 运算加速

举办目的

QuantLib C++开源项目自 2000/11/21 首版释出以来，早已成为欧、美金工领率的武林盟主。这样的成果来自于三方面，首先，它的开源性质，使它的代码成为全世界的公共财。你不需要花任何的费用，便可取得跟国外顶尖金融系统同样质量的代码。其次，它有全世界顶级的专家参与开发了所有实用的金融理论模型与底层架构。在 QuantLib 中你既有基本的 Black-Scholes 固定波动率(CV)模型，也有 Dupire 局部波动率(LV)模型、Heston 的随机波动率(SV)模型，以及最复杂的随机局部波动率(SLV)模型。另外，它大量使用先进的设计方法，如 Observer/Observable、Lazy Object、Visitor 等样式，让系统达到实时运算的效果。最后，它的架构完全符合银行金融计算实务作业的要求。QuantLib 中的金融计算与银行实务完全吻合，它的产出结果达到专业要求的精准度，这不是一般代码能达到的水平。

然而，由于 C++语言的进入门槛较高，再加上近来 Python 语言在 AI 与大数据领域的成功，愈来愈多从事金融量化工作的人士，选用 Python 成为他们的首选工具。有鉴于此，QuantLib 团队便以 SWIG 这个第三方开源工具，将 C++ DLL 库包覆起来，产生一个 Python 的接口，让我们去叫用它。透过此法，还可以产生其他语言的接口，如 Java 与 C#。SWIG 是一个很强大的中介软件，透过它我们可以轻松的由其他语言叫用 QuantLib C++的 DLL。由于它们的接口都很相似，因此学会一种语言的叫用，就等于其他语言也都会了。这其实是一鱼多吃的好处。希望所有需要金融计算的朋友，不要错过这个课程。因为不管你用甚么语言，都可以利用这个免费强大的 QuantLib 库，搭上欧、美专家提供的顶尖工具，提升你的生产力。

也因为 QuantLib 软件的开源性质，使得它的说明文件与学习资源并不普及。造成它只在专业人士之间流传使用，一般的金融行业工作者无法一亲芳泽。讲师本身在银行内部从事金融工程开发近 30 年，基于工作需要学习、使用 QuantLib 近 20 年。并使用 QuantLib 在银行内不开发全行资产负债管理系统，外汇结构商品 GPU 高速评价系统，以及股指期货 SABR 模型 GPU 高速套利系统。讲师在台湾证券公司、银行、金控担任过首席风险管近 15 年，并担任过银行交易室研发单位与结构商品单位主管，实务工作经验丰富。讲师拥有台大电机系本科学士与台湾中央大学财务博士学历，台湾证券分析人员资格，并担任台大、台湾师大、台湾科技大学兼任教授。

本课程分为四个单元，第一单元说明 QuantLib 的基础设施，包含时间相关类别、利率类别、利率期限结构类别、波动率期限结构类别，并以 Black-Scholes 模型为对象，实作香草型期权与奇异期权。包含二元(Binary)、亚洲式(Average)、回顾(Lookback)、障碍(Barrier)、双重障碍(Double Barrier)，等奇异期权。课程会介绍评价的各类数值方法，解析解、二叉树、有限差分、与模拟法。

第二单元将说明 QuantLib 中，期权的进阶定价模型，包含 Dupire 局部波动率模型、Heston 与 SABR 的随机波动率模型。对于这类复杂的模型，通常只有香草型选择权存在有解析解，因此数值方法的使用成为重点。Python 本身的数值计算能力并不是其强项，但是可以透过 DLL 叫用的方式，使用由 C/C++ 开发的高速计算核心。事实上，也可以将 CUDA 开发的计算核心，以 DLL 的方式，提供给 Python 使用。讲师会提供一个自行开发的结构商品模拟定价程序架构，让学员得以返回工作后，使用讲师的代码，完成实务工作的需求。

第三单元将说明如何把 GPU 高速运算功能与 QuantLib 结合起来。NVidia 的 CUDA 开发架构以成为 GPU 运算的主流标准。讲师将说明如何使用 PyCUDA 下，把 Python/CUDA C 整合为一，开发出一个高速的金融计算系统。课程中会介绍 GPU 的开发方式，并把评价代码值入 GPU 中。透过 PyCUDA 我们可以由 Python 呼叫 GPU 进行运算。

第四单元将说明如何把 GPU 高速运算功能与 QuantLib 结合起来，使用 Python 开发一个股指期权的高速运算交易系统。讲师将说明如何使用 PyCUDA 套件，把 Python/CUDA C 整合为一。我们会开发一个可以由交易所，实时接收市场股指期权报价的 Python 交易系统。此系统可以实时的计算选择权的波动性，以及 Delta、Gamma、Vega、Theta 等风险参数。并使用 SABR 模型来描绘 Volatility Smile，进一步计算出 SABR 模型的期权理论价格。投资人可以根据这些参数，作成买卖的下单决策，并由系统快速的下单到交易所。这个可客制化的 Python 交易系统原型，可以让学员进一步自行加工，作为自己的独有交易系统。

课程使用 Visual Studio 2022 做为 CUDA 程序的开发工具，并以 Windows 10 作为程序开发的环境。学员可以将程序代码携回，解决实务上的评价问题。程序代码没有使用的限制。

课程特色

- (一) 本课程讲师从事金融技术工作近 30 年的资历，曾在银行内部带领研发团队使用 QuantLib 链接库，开发银行的资产负债管理(ALM)系统，对银行五千亿人民币的部位，进行 MTM 的计算。另外，使用 QuantLib 与 GPU 绘图卡，开发高速运算的结构商品交易与风险控管系统。最近也使用 GPU 搭配 SABR 模型，完成股指期货期权高频套利系统的开发。
- (二) 本课程采计算机实机操作教学，请自行准备符合开发要求的计算机参加。开发系统为 Windows 10 或 11。参加第三单元的学生，计算机需有安插 NVIDIA 独显卡，如 GTX1660 或以上显卡，可执行 CUDA 12，并具备基础 C++程序开发的能力。

课程日期

期权基础开发班：2024 年 4 月 1 日，约 32 小时，含 4 小时在线讨论课程。
期权进阶模型班：2024 年 4 月 16 日，约 30 小时，含 4 小时在线讨论课程。
GPU 期权开发班：2024 年 5 月 1 日，约 30 小时，含 4 小时在线讨论课程。
股指期货交易系统开发班：2024 年 6 月 1 日，约 24 小时，含 4 小时在线讨论课程。

※以上时数为预估值，正确时间长度于课程上线方可确定。

主办单位：优量在线

课程地点：网络课程，录制影片搭配在线讨论课程

课程等级：涵盖基础与进阶课程

讲座介绍

董梦云 讲座

现 职：台湾金融研训院 2023 菁英讲座、台湾大学财金所兼任教授、昀腾科技技术长

学 历：台湾大学电机工程学系本科学士、台湾中央大学财务金融学系管理学博士

经 历：台湾永丰银行结构商品开发部部门总经理、台湾永丰金控首席风管官、台湾中华开发金控首席风管官、台湾凯基证券亚洲区首席风管官、台湾中国信托银行交易室研发科主管

专 长：风险管理理论与实务、金融工程、结构型商品设计与避险、GPU 程序设计、C#、C++/C、CUDA、OpenCL、Python、QuantLib。

参加对象

- (一) 各金融机构〈含银行、证券、保险等〉股票、外汇、商品、金融科技与市场风险相关部门之业务人员、财工人员；
- (二) 各金融机构〈含银行、证券、保险等〉信息人员；
- (三) 硕士以上金工专业的在职学生与业界人士。

课程费用

【期权基础开发班】每人每课程 RMB 4,500 元整

【期权进阶模型班】每人每课程 RMB 6,500 元整

【GPU 期权定价班】每人每课程 RMB 10,500 元整

【股指期权交易系统开发班】每人每课程 RMB 15,500 元整

※套装优惠：同时报名【期权基础开发班】+【期权进阶模型班】，RMB 10,000 元整。

※套装优惠：同时报名【期权基础开发班】+【期权进阶模型班】+【GPU 期权定价班】，
RMB 20,000 元整。

※课程费用包含影片收看，与课程范例代码，不提供讲义电子文件也不提供纸本的书面讲义。

报名方式

自即日起至开课前一周，可依据下列方式报名。

(一) 银行转账：

中国工商银行，卡号：[621226 200200 9788120](#)，姓名：董梦云

开户网点：莲花支行

请提供相关信息：手机号码，E-Mail，WeChatID 等

(二) 加微信好友转账：

WeCahtID：[andydong3137](#)

讲师连络方式：WeChat：[andydong3137](#)

手机：[15089192872](#)

E-mail：[andydong1209@gmail.com](#)

课程内容 ※本公司保留变更本课程内容之权利，相关异动以正式课表为准。

【期权基础开发班-33hrs】

课程名称	课程纲要	实作范例	时数
QuantLib 的安装与使用	QuantLib 链接库的由来 QuantLib 链接库的下载 QuantLib 链接库的内容 QuantLib-Python Package 下载与安装 QuantLib Package 使用测试	有 有	4 在线课程
日期与时间相关类别	Date 对象 Calendar 对象 Day Counter 对象 Period 对象 Frequency 列举 Schedules 对象	有 有 有 有 有 有	4
基础设施与利率、内插类别	Settings 对象 Quotes 对象 Interest Rate 对象 Interpolation 对象	有 有 有 有	2
利率期限结构类别	各类利率定义 Flat Forward 对象 Yield Term Structure 对象重要方法 其他 Yield Term Structure 对象 Yield Term Structure Handle 对象 内插利率曲线对象	有 有 有 有 有 有	4

波动率期限结构类别	Black Constant Vol.对象 Black Variance Curve 对象 Black Variance Surface 对象 Black Vol. Term Structure Handle 对象	有 有 有 有	2
期权产品相关类别	Exercise 对象 Payoffs 对象 Option 单资产对象 Option 多资产对象 Instrument 对象	有 有 有 有	4
解析解定价引擎类别	评价模型与 Processes 对象 常用简单计算器对象 GBS 模型 Pricing Engine 对象 隐含波动率计算 计算结果的产出	有 有 有 有 有	5
数值方法引擎类别	Binomial 计算引擎 有限差分计算引擎 Monte-Carlo 计算引擎	有 有 有	4
二元期权 组合期权	产品说明 市场数据处理 评价引擎调用	完整范例	1.5
平均期权 回顾期权	产品说明 市场数据处理 评价引擎调用	完整范例	2
障碍期权 双重障碍期权	产品说明 市场数据处理 评价引擎调用	完整范例	1.5

【期权进阶模型班-28hrs】

课程名称	课程纲要	实作范例	时数
Dupire 局部波动率模型	隐含波动率的计算 期权波动性曲面 局部波动率公式 使用 QuantLib 建构局部波动率曲面 期权定价实作范例	完整范例	4 在线课程
Heston 随机波动率模型	随机波动率模型 使用 QuantLib 定价引擎评价 Heston 随机过程的模拟法 模型参数市场校正	完整范例	6
SABR 随机波动率模型	SABR 模型说明 使用 QuantLib 定价引擎评价 模型参数市场校正 SABR 模型 Greeks 计算说明	完整范例	6
Python 的 DLL 调用与 C++ DLL 开发解析引擎	Python 的 C++ DLL 调用介绍 开发 BS 模型 DLL 解析解定价引擎 开发 SABR 模型 DLL 解析解定价引擎	完整范例	6
C++ DLL 开发模拟引擎	开发 BS 模型 DLL 模拟定价引擎 开发 Heston 模型 DLL 模拟定价引擎 雪球结构商品评价模拟定价应用	完整范例	6

【GPU 期权定价班-30hrs】

课程名称	课程纲要	实作范例	时数
GPU 与 CUDA 介绍	硬件的观点 多线程 软件的观点 CUDA 编程	完整范例	6 (含在线课程 4 小时)
CUDA 的变量与内存管理	内存 变数 缓存器 共享内存	完整范例	3
CUDA 的控制流程与同步	Warp 分歧 同步 Warp 投票 原子操作 全局原子锁	完整范例	3
PyCUDA 使用说明	PyCUDA 介绍与安装 使用 PyCUDA 进行 GPU 运算 PyCUDA 调用 CUDA 的 PTX 档 C#与 CUDA 的整合应用	完整范例	5
CUDA 下的模拟与 cuRand 链接库	CUDA 链接库 cuRand 链接库 其他链接库 7 个小矮人 可用工具	完整范例	5

GPU 版的 BS 定价 公式实作	BS 模型 Kernel 函数程序开发 CPU 端 PyCUDA 布建 CUDA 程序的执行 整合 Windows 程序的开发	完整范例	3
GPU 版的 BS 模型 模拟定价实作	自行开发 BS 模型模拟程序 模拟程序 Kernel 函数程序开发 CUDA 程序的执行 整合 Windows 程序的开发	完整范例	5

【股指期货交易系统开发班-24hrs】

课程名称	课程纲要	实作范例	时数
一、系统整体架构介绍	接收信息 处理信息 下单处理		1 在线课程
二、市场信息接收单元说明	交易所行情格式、内容及策略说明 商品代码拆解范例说明 串接行情 API 简易 GUI (pyQt)	完整范例	6 (含在线课程 3 小时)
三、SABR 模型运算说明	隐含波动性计算 SABR 模型参数校正 SABR 模型理论价格计算 SABR 模型 Greeks 计算说明	完整范例	6
四、交易策略范例说明	动态套利策略 Option Income Strategies Other Strategies	完整范例	5
五、程序下单模块说明	下单函式说明 保证金计算范例说明 串接下单 API 整合 GUI 完成简单的交易系统程序	完整范例	6