Quality and performance for a bundle of bits

Code production, maintenance and performance may be the unglamorous back end of the workflow, but how they are (mis)handled can have major impact

Today's financial engineers come from a wide range of backgrounds, such as computational finance, quantum field theory, fluid mechanics, or even geophysics. Whether their strength lies in numerical methods or continuum modeling, you can bet they didn't get into finance dreaming of delivering executables for derivatives pricing codes.

Traditionally, a team must first determine a computational approach for pricing a financial instrument, say a complex structured product. They then must implement and validate the chosen approach. Code, once written and tested, must also be maintained, as products, models and computational platforms evolve. Maintenance is not just a problem for code authors; it is compounded for management by the migration of top talent.

Your mind plus AI

How can you leverage quants' modeling skills to deliver quality, documented code that performs? For the last decade, SciComp Inc. has been tackling this problem, and its users, who include Merrill Lynch, Bank of America, ABN AMRO, Wachovia, and many other investment banks, have benefited. SciComp's products are based on an area of artificial intelligence called "software synthesis" - software that writes software. SciComp's SciFinance® system is designed to leverage quants' expertise, supercharging it to bring custom, complex derivatives to market in a fraction of the time previously required.

"SciFinance is a knowledge-based expert system that functions as a programming assistant. You can think of it as an intelligent compiler that codifies state-of-the-art mathematical and programming knowledge into a system of objects and rules used to generate optimized source code. This formalized expertise frees quants from the burden of programming details and enables them to focus on the modeling decisions that enhance their banks' profits." —Elaine Kant, President, SciComp Inc.

Using keywords, macros and mathematical equations, quants can specify at a high level the financial engineering problem they are tasked to implement. This higher-level language enables quants to focus on the modeling issues and to think about their modeling problems in terms natural to finance, rather than becoming mired in the details of a programming language or in shoe-horning existing modules into unanticipated configurations. From this higher-level specification, often only about half a page for even the most complex structured products, SciFinance generates a complete source code. Producing wrappers for Excel, .Net, COM and Java requires only the addition of one keyword.

Playing your instrument

SciFinance's flexibility allows it to cover the breadth of financial markets, including credit, convertible bonds, equity, energy, fixed income and foreign exchange markets, because its specifications are structured in terms of models (stochastic differential equations [SDEs], partial differential equations [PDEs] and partial-integro-differential equations [PIDES]) and associated numerical algorithms. This scope allows play by two types of users; generalists (such as risk groups interested in broad classes of financial instruments) and individual desks tasked with specific markets.

Tim Klassen, Head of equity-linked products quantitative research at Wachovia. "We had to write a modeling library from scratch to support the liquid and structured products businesses that Wachovia is building. Given the maturity of PDE and Monte Carlo pricing methodologies and the expertise of SciFinance in these areas we decided to use SciPDE and SciMC both for checking our own code and for production pricing of certain products. This allows our small team of quants to spend more time on the much less mature problems of coming up with algorithms for robust vol surface generation as well as the calibration of local volatilities and models with stochastic volatility and jumps, while at the same time providing quick turn-around for new products for the desk. As an example, within a few hours we were able to start exploring Napoleon pricing within a stochastic vol model with jumps."

SciFinance's flexibility also extends deeply into markets. All quants face different problems, and SciFinance gives them the freedom to make their preferred modeling decisions. The requirements may vary in data source available, maturity of the instrument and use of the instrument within a book. SciFinance supports both types of users through keywords and macros that allow them to customize the pricing implementation to very fine resolution according to the requirements.

As a specific example, in the case of credit derivatives, consider the different purposes of first to default (FTD) baskets. Some hedge FTD against equity tranches; others hedge FTD against portfolios of credit sensitive single name instruments. In either case, a practitioner may (or may not) require calibration of a first to default basket to term structures of survival probabilities or to term structures of credit default swap quotes. With simple keywords that make sense in credit instruments, a practitioner can specify these requirements in SciFinance. Similarly, quants concerned with convertible bonds may use some of the same credit keywords, and additionally, equity constructs such as local or stochastic volatility to capture skew.

With simple keywords that make sense in credit instruments, a practitioner can specify these requirements in SciFinance

What's the learning curve?

Studies have shown that programming time is roughly proportional to lines of code handwritten, even in the new object-oriented programming paradigms. SciFinance specifications are typically orders of magnitude more concise than the codes that are generated. Thus a specification of a dozen lines can be quickly written but may "compile" into thousands of lines of source code, in minutes. Specification keywords and macros are higher-level constructs that have self evident meaning; for example: BlackScholes, CIR, ADI, MonteCarlo, ImportanceSample, and ExcelWrapper. While some investment in time is then required to learn the high-level language and its composition, an extensive catalog of SciFinance example specifications is provided to expedite the process. Typically, first time practitioners can produce custom standalone executables within hours of installation of SciFinance.

I want my PDE

Financial engineering considers mostly diffusion problems ranging from one spatial dimension to hundreds. Free boundary problems are ubiquitous in the form of early exercise and continuous call and/or conversion features.

Numerical PDE solutions are optimal for many low dimensional problems. Implementations in code

SciFinance®

Say you need to price the fair spreads on a three-tranche collateralized debt obligation. The specification for this is 8 lines, not including comments, and specifies computation of the fair spreads of a three tranched CDO using Li's method with computation of Greek sensitivities, deterministic discounting, and quasi Monte Carlo. The specification invokes 35 lines of macros and produces 1200 lines of code in two minutes. The specification can be customized to hundreds of variations in the waterfall, tranching, default modeling, default statistics, default triggers, calibration parameters, Greek sensitivities, etc.

```
(* mcCSO5.s *)
(* n-tranche CDO using Li's method with Greeks *)
MonteCarlo;
(* Default Model*)
ReducedFormLi;
(* Interest Rate Model *)
DeterministicShortRate;
(* Instrument Structure *)
WaterFall FairSpreads[nTranche->3];
(* Greeks *)
Delta == der[SeniorPremium, h];
(* Numerical Methods *)
Sobol;
(* I/O *)
ReadArray[CouponDates, nCoupon];
Output[{Premium[i], Loss[i], Spread[i]}, {i,1,3}];
Output[Delta, "Delta.out"];
```

Figure 1: Example SciFinance specification. Generates 1200 lines of source code.

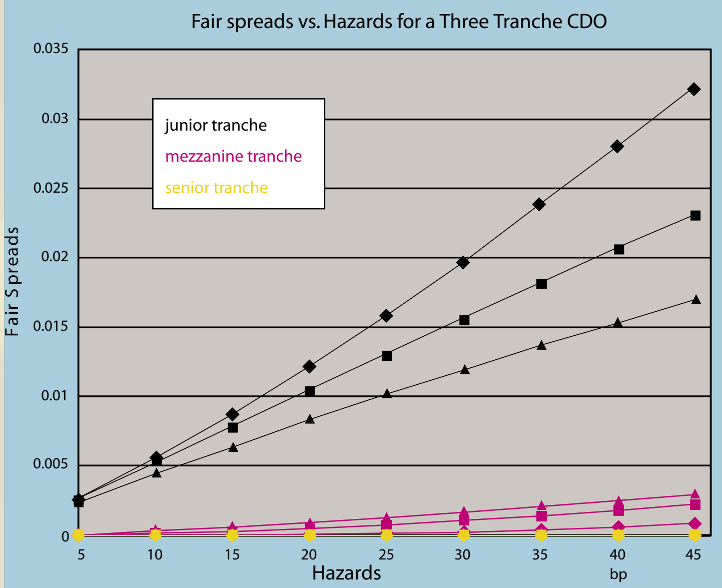

Figure 2: Shows the fair spreads of the three tranches for various parametric values of the correlation: The ◆ curve has no correlation, while the ■ and ▲ curves are exponential correlations of 0.25 and 0.5 respectively. 10,000 paths were used. All three curves for the senior tranche lie along the axis at this scale.

This abstraction makes it easier to add new methods, to mix and match existing methods, and to optimize each set of selections for any given platform

are not difficult to generate, unless of course, one requires efficiency, unconditional stability, accuracy and monotonic convergence. To create robust production PDE code, the generic methods discussed in an introductory numerical methods class simply will not suffice. Specialized linear and non-linear system solvers, time and spatial discretization schemes, and grid generation algorithms are required. SciFinance makes these algorithms easily accessible via keywords.

"Expert numerical methods are researched or plucked from the literature and integrated into SciFinance on a continuing basis, so that our customers have access to the latest techniques. This knowledge is further encapsulated into keywords, so that they are easily accessible. Customization is easy. Customers can add their own macros for equations or algorithms that are often used, specifications can call other specifications, etc." said Curt Randall, Executive VP for SciComp.

MC for me

Advanced Monte Carlo and calibration methods are also available to address the spectrum of problems that must be efficiently implemented in production systems. Brownian bridge continuity corrections for discrete sampling of extrema, high order discretization for complex (multi-dimensional) processes, a range of variance reduction techniques, and Longstaff-Schwartz regression with fully customizable basis sets are among capabilities available via keyword. Methods from Levenberg Marquardt and simulated annealing to simple bisection, can be applied to calibration problems.

Easy availability of distinct methods removes the barrier (programming) to cross checking of models. "For example, consider a structured product with forward starting features. One might chose a stochastic volatility model, either proprietary equations or selected from the literature (e.g., Heston's model). Using SciFinance both PDE and MC models can be synthesized and cross checked in hours. The PDE model might use an advanced solver like ADI while the MC model might use a high order discretization such as Milstein. Both are invoked by a single keyword," added Randall.

Keeping IT real

Synthesized SciFinance codes are well documented, structured and require no C-level maintenance from practitioners. Changes are made at the concise specification level and the codes are regenerated, making global and local optimizations for new specification. "Leaving aside the fact that the generated code is sophisticated, from a management perspective the code is effectively self-documenting in the sense that quants can easily understand a specification created by another person." Says Jim Gatheral, Head of Equity Markets Quantitative Analytics at Merrill Lynch.

"Our convertible bond model was created using SciFinance." Gatheral explains. "Although the SciPDE specification is naturally very long and the interface with our risk management system still rather complex, SciFinance makes it much easier to update the code to accommodate new features. I am convinced that SciFinance made the process of implementing the model much more efficient than it would have been otherwise."

SciComp offers a variety of methods for automatically integrating SciFinance synthesized codes into production environments, including .NET, Java and COM wrappers, Microsoft Excel add-ins and a variety of I/O and error handling regimes. SciComp can also customize integration with in-house risk management and trading systems.

Building a better library

Libraries are a fact of life in financial engineering; either in the form of legacy libraries built in house or purchased from vendors (including SciComp with a different hat). A fundamental difficulty with most libraries is their inflexibility in the face of new methods, new classes of instruments, and new platforms. In a sense, SciFinance itself is also a library, but it is a library of mathematical and programming methods at a much more abstract level. This abstraction makes it easier to add new methods, to mix and match existing methods, and to optimize each specific set of selections for any given platform.

With SciFinance, quants can build and modify custom libraries, which like any software system (built or purchased) includes the executables used day-to-day in a production system. But now, if a change is needed to respond to a new direction in the market, it is a simple task to customize the specification (rather than the source code) of the existing production library element. SciFinance can then generate the executable for the revised specification. Quants can respond to market-driven changes themselves, without waiting for updates from a library vendor or IT department, or struggling with old in-house code whose authors are no longer present.

When customers don't have the time or quant resources to develop a complex model from scratch, SciComp can quickly offer standalone pricing models that integrate with a library, trading or risk management system. Since these models are developed with SciFinance, they share the same flexibility in terms of ease of adding new methods, optimization and customization.

Don't sweat the code stuff

Now that the problems of code production, maintenance, and performance have been vanquished, quants are free to respond to the growing derivatives market, putting new spins on old problems and relishing the challenges of new ones. CDO^3 anyone?

SciComp Inc.
5806 Mesa Drive, Suite 250
Austin, Texas USA 78731
+1 (512) 451-1050
www.scicomp.com

W