Design of Problem-Solving Environment for Contingent Claim Valuation

F. Oliver Bunnin, Yike Guo, and John Darlington

Department of Computing, Imperial College
London, SW7 2BZ, England

Abstract. We present the design and initial implementation of a problem-solving environment that values industrial investment projects. We use the contingent claim, or real option, valuation method, which views a project as a claim to future cash flows which are dependent on underlying stochastic factors, such as the market price of a produced commodity. The problem-solving environment enables the user to use domain and mathematical level constructs to specify the nature of the project and the stochastic behaviour of underlying factors. Meaning preserving symbolic rewrite rules transform these specifications into problem representations that are suitable for numerical solution. The transformed problem representations are then combined with components implementing parallel algorithms in order to compute solutions.
The aim of this approach is to benefit strategic industrial decision-making by enabling high-level and flexible problem formulation and by using high performance computational resources.

1 Introduction

Contingent claim modelling, or the Real Option method, is the use of methods from stochastic mathematics to value contracts or investments whose value is contingent on the uncertain future state of the world. It is regarded as one of the most general and sophisticated investment valuation methods applicable to a broad class of industrial investment projects. These methods were pioneered in the financial field, for the valuation and risk management of financial derivatives, but are now becoming popular in a wide range of other industrial contexts [1,10]. Prime examples are investment decision making for oil and gas wells and for the development and testing of new medicines in the pharmaceutical industry [10].

The basic method is to model the system on which the claim is contingent as a stochastic process and then find a functional relationship between the value of the claim and the state of the underlying system throughout a given period of time. This involves the use of stochastic and partial differential equations, dynamic programming and optimal stopping theory. Suitable numerical methods to solve these problems are finite differences, lattice methods and Monte Carlo simulation [11]. There exists a vast literature on these techniques, both mathematical and numerical, yet generating an efficient method for a particular

R. Sakellariou et al. (Eds.): Euro-Par 2001, LNCS 2150, pp. 935–938, 2001.

problem, on a particular architecture, remains a serious challenge, due to the computational and software complexity.

There are significant technical and practical challenges to the computer implementation of this methodology. The algorithms required for contingent modelling are computationally intensive, involving the solution of high dimensional partial differential equations and/or stochastic simulations. Efficient algorithms must be selected and implemented, on a problem by problem basis. The cost of employing expert mathematicians with first class programming skills for model implementation is very high and increases software development costs. Auditing and validation of contingent claims systems are mandatory, but notoriously difficult when programs are hand-coded and dynamically updated.

Therefore the overall objective is to facilitate rapid development of transparent and theoretically sound investment project valuation programs delivering real time results using advanced hardware, algorithms and software techniques. This is achieved by allowing users to specify models naturally and easily at the mathematical level and by using symbolic rewrite rules and component technology to automatically construct solution code.

2 Overview of Environment

In order to simplify and structure the generation of solution code from specifications we generate components representing views of the problem at different levels of abstraction: the domain level, the mathematical level and the numerical level. Such views are often used in PSEs: for example see Gallopoulos [2]. At the domain level the user specifies the project in terms of its contingent cash flows. These define the profit rate function $\Pi(t, d, X)$ which is a function of t, time, d, the decision variables of the project management, and X the random underlying factors. Components of d may be the capacity of machinery employed, whether or not to suspend production and so on. Components of X may be the market price of output or input factors, or macroeconomic variables that influence market prices.

We assume the project has a finite horizon, T. The objective of the management is then to maximise the expected profit, discounted over time, of the project over its lifetime:

$$\max_{d \in D} E[\int_0^T e^{-rs} \Pi(s, d, X) ds] \tag{1}$$

where D is the space of allowable decisions, r is an appropriate continuously compounded interest rate and expectation is taken using a risk-neutral probability measure. Alternative utility based formulations are possible when the cash flows of the project cannot be replicated by trading in market instruments, i.e. when the market is incomplete (see Henderson and Hobson [3]). This is a stochastic control problem, and its relation to the non-linear Hamilton-Jacobi-Bellman PDE is discussed in Oksendal [9]. The mathematical treatment of various formulations of contingent claim problems are discussed in Lund and Oksendal [8].

We construct a C++ component to represent the project, which has as its data members the specified decision variables d, the random underlying factors X and as a member function the profit function Π. Note that at this point we have not yet defined the stochastic behaviour of X. This is because often the user may wish to try out various different stochastic models, and over time may change his or her view of which model is most appropriate. The domain level C++ component is independent of model, and thus need not change when the model or its numerical implementation changes.

A stochastic model for X is specified using stochastic differential equations (SDEs). For this we have defined notation in Mathematica. Once the SDE is represented in this way we can apply symbolic transformations that we have written in Mathematica. The symbolic transformations encode results from stochastic analysis, such as the Ito formula and Girsanov's theorem [9], and results from contingent claim modelling theory. Thus we automatically formulate the stochastic control problem defined in equation (1) with the project defined profit function and model defined SDEs substituted appropriately. A second set of symbolic rewrite rules are used to approximate the problem for numerical solution. For example the continuous time SDEs may be discretised in order to be used in a Monte Carlo simulation.

The symbolic transformations applied to the SDE model result in a C++ model component. This is combined with the C++ project component and with predefined parallel algorithm components to form a computational component solving the specific problem. The algorithm components are designed to abstract common patterns of parallel behaviour in stochastic modelling, such as time stepping over a distributed spatial grid, updating discretely sampled variables across processors, combining results from independent paths of a stochastic variable calculated on separate processors. As such they support a wide class of algorithms when combined with sequential algorithms, such as linear system solvers and random number generators. This facilitates the rapid construction of new algorithms, such as the Pseudospectral method, by re-using existing components. The code formed by composing algorithm components and solvers is re-usable in other applications. Note that the re-use of financial algorithmic code is widespread in practise: perhaps the best example is the use of sophisticated methods for constructing discount factor curves from liquid traded instruments, which are used in almost every derivative valuation.

3 Conclusion

Contingent claim valuation is an important commercial area that presents computationally intensive and mathematically interesting problems. It is an area that is eminently suitable for a problem-solving environment which generates software that utilises high performance computers and allows end-users to compute solutions from high level specifications. Related work by Kant et al. [6,7] and Pantazopoulos and Houstis [5] has concentrated on financial contingent claims and finite difference methods. We build on such research, focussing on the more general and more complicated area of industrial project valuation, and utilising a wider range of solution techniques. This paper presents an overview of our design and initial implementation and indicates the direction of further research.

We have implemented a set of symbolic transformations and using these have constructed several stochastic model components. Components for parallel simulation and sequential finite difference and pseudospectral algorithms have also been built. We have demonstrated the effectiveness of the design through the construction of computational components for exotic financial contingent claims and are presently working on extension to more complex industrial project valuation problems. For more material on this research please see `http://www.doc.ic.ac.uk/~fob1/project.html` .

Acknowledgments

This work is supported by the EPSRC grant GR N14675.

References

1. Dixit, Avinash K., and Robert S. Pindyck, Investment under Uncertainty. Princeton University Press, 1994 935
2. Gallopoulos E., E. Houstis, J. Rice, Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, IEEE Computational Science and Engineering vol. 1 no. 2 (1994) 936
3. Henderson, V. and Hobson, D. Real Options with Constant Relative Risk Aversion, preprint, University of Warwick 936
4. Hughston, Lane P., Zervos, Mihail, Martingale Approach to Real Options, preprint King's College London 2000
5. Pantazopoulos K. N., Houstis E. N., Modern software techniques in computational finance, in Modern Software Tools for Scientific Computing, Birkhauser, 1997 937
6. Kant E., Synthesis of Mathematical Modelling Software, IEEE Software 1993 937
7. Kant E., C. Randall, A. Chhabra, Using Program Synthesis to Price Derivatives, Journal of Computational Finance vol. 1 no. 2 (Winter 1997/98) 97-129 937
8. Lund D. and Oksendal B. (eds), Stochastic models and option values : applications to resources, environment and investment problems (Amsterdam ; Oxford : North-Holland, 1991) 936
9. Oksendal B., Stochastic Differential Equations, (Springer, 1998) 936, 937
10. Trigeorgis L., Real Options: Managerial Flexibility and Strategy in Resource Allocation, MIT Press 1996 935
11. Wilmott, P., Dewynne, J., Howison, S., Option Pricing. Mathematical Models and Computation, Oxford Financial Press 1993 935