

FROM NAVIER-STOKES TO BLACK-SCHOLES: NUMERICAL METHODS IN COMPUTATIONAL FINANCE

DANIEL J. DUFFY

ABSTRACT. In this article we give a general overview of the numerical methods (in particular the finite difference method) to approximate the partial differential equations that describe the behaviour of financial products (such as stocks, options, commodities and interest rate products). These products are traded in the marketplace and it is important to price them using accurate and efficient algorithms. Furthermore, financial institutions need to compute and monitor the risks associated with these financial instruments and portfolios of these instruments.

The focus in this article is to trace the emergence of advanced numerical techniques and their applications to computational finance during the last twenty-five years. It is aimed at a mathematical audience with a passing acquaintance of partial differential equations (PDEs) and finite difference methods. In particular, time-dependent convection-diffusion-reaction PDEs will take centre-stage because they model a wide range of financial products.

1. A SHORT HISTORY OF COMPUTATIONAL FINANCE

Computational Finance can be defined as a set of mathematical and engineering techniques to solve complex problems in finance. It has grown steadily during the last thirty years as financial services became global and computing power increased exponentially. When the Cold War ended the market had acquired access to a large pool of physicists, mathematicians and computer scientists, or quants as they became known on Wall Street. These quants applied their knowledge to solve complex derivatives pricing problems. Growth was explosive until the financial crash of 2007/2008. After the crash

2010 *Mathematics Subject Classification.* 91G60, 65C30.

Key words and phrases. Financial Mathematics, Options, Derivatives, Computation, PDE.

Received on 20-2-2015; revised 8-3-2015.

many of the exotic structured products that these quants had invented were abandoned in favour of simpler ones. The events of the last seven years have certainly proved that finance is not physics and that models are, after all, just models of reality and not reality itself.

2. COMPUTATIONAL FINANCE 101: PLAIN CALL OPTIONS

Before we jump into the mathematics and numerical analysis of partial differential equations we try to sketch the financial context in which they are used. It is impossible to discuss the context in any great detail and we refer the reader to Wilmott [11]. It is written in a style that should appeal to mathematicians.

In order to reduce the scope we focus exclusively on the most fundamental of all financial instruments, namely equity (also known as stock or shares). Holding equity means that you own part of a company. If the company goes bankrupt the value of your shares is effectively zero or thereabouts. In short, you have lost your investment! The investor paid up front and she was probably expecting the share price to increase in the future. Most people are optimists and hence they buy shares in the hope that they will rise in price. But this is risky because if the share price drops they will make a loss. There is however, a less risky approach. Let us assume that you expect the share price of the ABC company to rise from \$100 to \$140 in the next three months (for example, you consulted your crystal ball on this and that is what it told you). So, instead of buying the share for \$100 now you might like to have the option to wait for three months and then buy the share. You can then buy a *call option* that gives you the right but not the obligation to buy the share three months into the future for a certain *strike price*. Of course, having the right but not the obligation to buy a share at some time in the future comes at a price and this must be paid by the investor up-front. For example, you can buy a call option with strike price \$120 that expires in three months time. If the price is greater than \$120 then you have made a profit. If the price is less than \$120 at expiration the option is worthless and you have lost your initial investment.

We have not addressed the issue of how to compute the option price. This is precisely the famous Black Scholes formula [1] that allows us

to compute the option price analytically. In more complicated cases we need to resort to numerical methods as discussed in this article.

3. THE MATHEMATICS OF PDES IN COMPUTATIONAL FINANCE: HELICOPTER VIEW

In general, the PDEs of relevance are of the *convection-diffusion-reaction* type in n space variables and one time variable. The space variables correspond to underlying financial quantities such as an asset or interest rate while the non-negative time variable t is bounded above by the *expiration* T . The space variables take values in their respective positive half-planes.

We model derivatives that are described by so-called initial boundary value problems of parabolic type [10]. To this end, consider the general parabolic equation:

$$Lu \equiv \sum_{i,j=1}^n a_{ij}(x,t) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^n b_j(x,t) \frac{\partial u}{\partial x_j} + c(x,t)u - \frac{\partial u}{\partial t} = f(x,t) \quad (1)$$

where the functions a_{ij}, b_j, c and f are real-valued $a_{ij} = a_{ji}$, and

$$\sum_{i,j=1}^n a_{ij}(x,t) \alpha_i \alpha_j > 0 \quad \text{if} \quad \sum_{j=1}^n \alpha_j^2 > 0. \quad (2)$$

In equation (2) the variable x is a point in n -dimensional space and t is considered to be a positive time variable. Equation (1) is the general equation that describes the behaviour of many derivative types. For example, in the one-dimensional case ($n = 1$) it reduces to the famous Black-Scholes equation (Here $t^* = T - t$):

$$\frac{\partial V}{\partial t^*} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D)S \frac{\partial V}{\partial S} - rV = 0 \quad (3)$$

where V is the derivative type (for example a call or put option), S is the underlying asset (or stock), σ is the constant volatility, r is the interest rate and D is a dividend. Equation (3) is a special case and it can be generalised to include more general kinds of options.

Equation (3) can be generalised to the multivariate case:

$$\frac{\partial V}{\partial t^*} + \sum_{j=1}^n (r - D_j) S_j \frac{\partial V}{\partial S_j} + \frac{1}{2} \sum_{i,j=1}^n \rho_{ij} \sigma_i \sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} = rV. \quad (4)$$

This equation models a multi-asset environment. In this case σ_i is the volatility of the i^{th} asset and ρ_{ij} is the correlation ($-1 \leq \rho_{ij} \leq 1$) between assets i and j . In this case we see that equation (4) is written as the sum of three terms:

- Interest earned on cash position

$$r \left(V - \sum_{j=1}^n S_j \frac{\partial V}{\partial S_j} \right). \quad (5)$$

- Gain from dividend yield

$$\sum_{j=1}^n D_j S_j \frac{\partial V}{\partial S_j}. \quad (6)$$

- Hedging costs or slippage

$$- \frac{1}{2} \sum_{i,j=1}^n \rho_{ij} \sigma_i \sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j}. \quad (7)$$

Our interest is in discovering robust numerical schemes that produce reliable and accurate results irrespective of the size of the parameter values in equation (4).

Equation (1) has an infinite number of solutions in general. In order to reduce this number to one, we need to define some constraints. To this end, we define so-called *initial condition* and *boundary conditions* for (1). We achieve this by defining the space in which equation (1) is assumed to be valid. In general, we note that there are three types of boundary conditions associated with equation (1) (see [10]). These are:

- First boundary value problem (Dirichlet problem).
- Second boundary value problem (Neumann, Robin problems).
- Cauchy problem.

The first boundary value problem is concerned with the solution of (1) in a domain $D = \Omega \times (0, T)$ where Ω is a bounded subset of \mathbb{R}^n and T is a positive number. In this case we seek a solution of (1) satisfying the conditions:

$$\begin{aligned} u|_{t=0} &= \varphi(x) && \text{(initial condition)} \\ u|_{\Gamma} &= \psi(x, t) && \text{(boundary condition)} \end{aligned} \quad (8)$$

where Γ is the boundary of Ω . The boundary conditions in (8) are called Dirichlet boundary conditions. These conditions arise when

we model single and double barrier options in the one-factor case (see [5]). They also occur when we model plain options.

The second boundary value problem is similar to (8) except that instead of giving the value of u on the boundary Γ the directional derivatives are included, as seen in the following specification:

$$\left(\frac{\partial u}{\partial \eta} + a(x, t)u\right)|_{\Gamma} = \psi(x, t). \quad (9)$$

In this case $a(x, t)$ and $\psi(x, t)$ are known functions of x and t , and $\frac{\partial}{\partial \eta}$ denotes the derivative of u with respect to the outward normal η at Γ . A special case of (9) is when $a(x, t) \equiv 0$; then (9) represents the *Neumann boundary conditions*. These occur when modelling certain kinds of put options. Finally, the solution of the Cauchy problem for (1) in the strip $\mathbb{R}^n \times (0, T)$ is given by the initial condition:

$$u|_{t=0} = \varphi(x) \quad (10)$$

where $\varphi(x)$ is a given continuous function and $u(x, t)$ is a function that satisfies (1) in $\mathbb{R}^n \times (0, T)$ and that satisfies the initial condition (10). This problem allows negative values of the components of the independent variable $x = (x_1, \dots, x_n)$. A special case of the Cauchy problem can be seen in the modelling of one-factor European and American options (see [11]) where x plays the role of the underlying asset S . Boundary conditions are given by values at $S = 0$ and $S = \infty$. For European options these conditions are:

$$\begin{aligned} C(0, t) &= 0 \\ C(S, t) &\rightarrow S \quad \text{as} \quad S \rightarrow \infty. \end{aligned} \quad (11)$$

Here C (the role played by u in equation (1)) is the variable representing the price of the call option. For European put options the boundary conditions are:

$$\begin{aligned} P(0, t) &= Ke^{-r(T-t^*)} \\ P(S, t) &\rightarrow 0 \quad \text{as} \quad S \rightarrow \infty. \end{aligned} \quad (12)$$

Here P (the role played by u in equation (1)) is the variable representing the price of the put option, K is the strike price, r is the risk-free interest rate, T is the expiration and t is the current time.

From this point on we assume the following ‘canonical’ form for the operator L in equation (1):

$$Lu \equiv -\frac{\partial u}{\partial t} + \sigma(x, t)\frac{\partial^2 u}{\partial x^2} + \mu(x, t)\frac{\partial u}{\partial x} + b(x, t)u = f(x, t) \quad (13)$$

where σ, μ, b and f are known functions of x and t .

We have given a global introduction to the kinds of linear partial differential equations that are used in computational finance. We are unable to discuss other topics such as nonlinear PDEs, free and moving-boundary value problems, qualitative properties of equation (1) (for example, criteria for existence and uniqueness of the solution of equation (1)) and applications to computational finance. For a discussion of these topics we refer the reader to [5].

For the rest of this article we restrict our attention to the linear one-factor PDE defined by equation (13) in conjunction with auxiliary conditions to ensure existence and uniqueness. We also assume that all the coefficients and inhomogeneous term in equation (13) are known.

4. THE FINITE DIFFERENCE METHOD (FDM) IN COMPUTATIONAL FINANCE

For completeness, we formulate the initial boundary value problem whose solution we wish to approximate using the finite difference method.

Define the interval $\Omega = (A, B)$ where A and B are two real numbers. Further let $T > 0$ and $D = \Omega \times (0, T)$.

The formal statement of the idealised problem is:

With

$$Lu \equiv -\frac{\partial u}{\partial t} + \sigma(x, t)\frac{\partial^2 u}{\partial x^2} + \mu(x, t)\frac{\partial u}{\partial x} + b(x, t)u,$$

find a function $u : D \rightarrow \mathbb{R}^1$ such that

$$Lu = f(x, t) \text{ in } D \tag{14}$$

$$u(x, 0) = \varphi(x), \quad x \in \Omega \tag{15}$$

$$u(A, t) = g_0(t), \quad u(B, t) = g_1(t), \quad t \in (0, T). \tag{16}$$

The initial-boundary value problem (14)-(16) is general and it subsumes specific cases from the option pricing literature (in particular it is a generalisation of the original Black-Scholes equation).

In general, the coefficients $\sigma(x, t)$ and $\mu(x, t)$ represent *volatility* (diffusivity) and *drift* (convection), respectively. Equation (14) is called the *convection-diffusion-reaction equation*. It serves as a model for many kinds of physical and economic phenomena. Much research

has been carried out in this area, both on the continuous problem and its discrete formulations (for example, using finite difference and finite element methods). In particular, research has shown that standard centred-difference schemes fail to approximate (14)-(16) properly in certain cases (see [4]) .

The essence of the finite difference method is to discretise equation (14) by defining so-called discrete mesh points and approximating the derivatives of the unknown solution of system (14) - (16) in some way at these mesh points. The eventual goal is to find accurate schemes that will be implemented in a programming language such C++ or C# for the benefit of traders and risk management. Some typical attention points are:

- The PDE being approximated may need to be pre-processed in some way, for example transforming it from one on a semi-infinite domain to one on a bounded domain.
- Determining which specific finite difference scheme(s) to use based on quality requirements such as accuracy, efficiency and maintainability.
- Essential difficulties to resolve: convection dominance, avoiding oscillations and how to handle discontinuous initial conditions, for example.
- Developing the algorithms and assembling the discrete system of equations prior to implementation.

Our goal is to approximate (14)-(16) by finite difference schemes. To this end, we divide the interval $[A, B]$ into the sub-intervals:

$$A = x_0 < x_1 < \dots < x_J = B$$

and we assume for convenience that the mesh-points $\{x_j\}_{j=0}^J$ are equidistant, that is:

$$x_j = x_{j-1} + h, \quad j = 1, \dots, J. \quad \left(h = \frac{B - A}{J} \right)$$

Furthermore, we divide the interval $[0, T]$ into N equal sub-intervals $0 = t_0 < t_1 < \dots < t_N = T$ where $t_n = t_{n-1} + k$, $n = 1, \dots, N$ ($k = T/N$).

(It is possible to define non-equidistant mesh-points in the x and t directions but doing so would complicate the mathematics and we would be in danger of losing focus).

The essence of the finite difference method lies in replacing the

derivatives in (14) by divided differences at the mesh-points (x_j, t_n) . We define the difference operators in the x -direction as follows:

$$\begin{aligned} D_+u_j &= (u_{j+1} - u_j)/h, & D_-u_j &= (u_j - u_{j-1})/h \\ D_0u_j &= (u_{j+1} - u_{j-1})/2h, & D_+D_-u_j &= (u_{j+1} - 2u_j + u_{j-1})/h^2. \end{aligned}$$

It can be shown by Taylor expansions that D_+ and D_- are first-order approximations to $\frac{\partial}{\partial x}$, respectively while D_0 is a second-order approximation to $\frac{\partial}{\partial x}$. Finally, D_+D_- is a second-order approximation to $\frac{\partial^2}{\partial x^2}$.

We also need to discretise the time dimension and to this end we consider the scalar initial value problem:

$$\begin{cases} Lu \equiv u'(t) + a(t)u(t) = f(t), \forall t \in [0, T] \\ \text{with } a(t) \geq \alpha > 0, \forall t \in [0, T]. \\ u(0) = A. \end{cases} \quad (17)$$

The interval where the solution of (17) is defined is $[0, T]$. When approximating the solution using finite difference equations we use a discrete set of points in $[0, T]$ where the discrete solution will be calculated. To this end, we divide $[0, T]$ into N equal intervals of length k where k is a positive number called the *step size*. In general all coefficients and discrete functions will be defined at these *mesh points*. We draw a distinction between those functions that are known at the mesh points and the solution of the corresponding difference scheme. We adopt the following notation:

$$\begin{aligned} a^n &= a(t_n), f^n = f(t_n) \\ a^{n,\theta} &= a(\theta t_n + (1 - \theta)t_{n+1}), 0 \leq \theta \leq 1, 0 \leq n \leq N - 1 \\ u^{n,\theta} &= \theta u^n + (1 - \theta)u^{n+1}, 0 \leq n \leq N - 1. \end{aligned} \quad (18)$$

Not only do we have to approximate functions at mesh point but we also have to come up with a scheme to approximate the derivative appearing in (17). There are several possibilities and they are based on *divided differences*. For example, the following divided differences

approximate the first derivative of u at the mesh point $t_n = n * k$;

$$\left. \begin{aligned} D_+ u^n &\equiv \frac{u^{n+1} - u^n}{k} \\ D_- u^n &\equiv \frac{u^n - u^{n-1}}{k} \\ D_0 u^n &\equiv \frac{u^{n+1} - u^{n-1}}{2k}. \end{aligned} \right\} \quad (19)$$

We now introduce a number of important and useful difference schemes that approximate the solution of (17). The main schemes are:

- Explicit Euler.
- Implicit Euler.
- Crank Nicolson (or box scheme).

The explicit Euler method is given by:

$$\begin{aligned} \frac{u^{n+1} - u^n}{k} + a^n u^n &= f^n, \quad n = 0, \dots, N - 1 \\ u^0 &= A \end{aligned} \quad (20)$$

whereas the implicit Euler method is given by:

$$\begin{aligned} \frac{u^{n+1} - u^n}{k} + a^{n+1} u^{n+1} &= f^{n+1}, \quad n = 0, \dots, N - 1 \\ u^0 &= A. \end{aligned} \quad (21)$$

Notice the difference: in (20) the solution at level $n + 1$ can be directly calculated in terms of the solution at level n while in (21) we must rearrange terms in order to calculate the solution at level $n + 1$. The next scheme is called the Crank-Nicolson or box scheme and it can be seen as an average of the explicit and implicit Euler schemes. It is given as:

$$\begin{aligned} \frac{u^{n+1} - u^n}{k} + a^{n, \frac{1}{2}} u^{n, \frac{1}{2}} &= f^{n, \frac{1}{2}}, \quad n = 0, \dots, N - 1 \\ u^o &= A \text{ where } u^{n, \frac{1}{2}} \equiv \frac{1}{2}(u^n + u^{n+1}). \end{aligned} \quad (22)$$

The discussion in this section has prepared us for a discussion of the Black-Scholes partial differential equation.

5. EXAMPLE: THE BLACK-SCHOLES PDE AND ITS APPROXIMATION

Probably one of the most famous formulae in computational finance is due to Fischer Black, Myron Scholes and Robert Merton [1]. It has become popular with traders to price and hedge (a hedge is a trade to reduce risk) options.

We introduce the generalised Black Scholes formula to calculate the price of a call option on some underlying asset. In general the call price is a function of six parameters:

$$C = C(S, K, T, r, \sigma, t) \quad (23)$$

where the parameters have the following meaning [8]:

- S = asset price.
- K = strike (exercise) price.
- T = exercise (maturity) date.
- r = risk-free interest rate.
- σ = constant volatility.
- b = cost of carry.

We can view the call option price C as a function that maps a vector of parameters into a real value. The exact formula for C is given by:

$$C = S e^{(b-r)T} N(d_1) - K e^{-rT} N(d_2) \quad (24)$$

where $N(x)$ is the standard cumulative normal (Gaussian) distribution function defined by

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-y^2/2} dy \quad (25)$$

and where

$$\begin{cases} d_1 = \frac{\ln(S/K) + (b + \sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 = \frac{\ln(S/K) + (b - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}. \end{cases} \quad (26)$$

The cost-of-carry parameter b has specific values depending on the kind of derivative security [8]:

- $b = r$, we have the Black and Scholes stock option model.
- $b = r - q$, the Morton model with continuous dividend yield q .
- $b = 0$, the Black futures option model.

- $b = r - R$, the Garman and Kohlhagen currency option model, where R is the foreign risk-free interest rate.

Thus, we can find the price of a plain call option by using formula (24).

The formula needs six input parameters, one of which (namely, the volatility) cannot be found from the market and then special methods must be employed to estimate it. A discussion of this problem is outside the scope of this article. Even though the assumptions upon which formula (24) are based do not hold in all practical cases (see Hull 2006 for a discussion) it is nonetheless the motivator for more general cases for which an analytical solution is not available. In these cases we must resort to numerical methods, for example using the finite difference method that approximates the so-called Black-Scholes PDE:

$$LV \equiv -\frac{\partial V}{\partial t} + \sigma(S, t) \frac{\partial^2 V}{\partial S^2} + \mu(S, t) \frac{\partial V}{\partial S} + b(S, t)V$$

where

$$(27)$$

$$\begin{aligned} \sigma(S, t) &= \frac{1}{2}\sigma^2 S^2 \\ \mu(S, t) &= rS \\ b(S, t) &= -r. \end{aligned}$$

The corresponding fitted scheme is now defined as:

$$\begin{aligned} L_k^h V_j^n &= -\frac{V_j^{n+1} - V_j^n}{k} + \rho_j^{n+1} D_+ D_- V_j^{n+1} + \mu_j^{n+1} D_0 V_j^{n+1} \\ &\quad + b_j^{n+1} V_j^{n+1}, \\ \text{for } 1 \leq j \leq J-1, &\text{ where} \\ \rho_j^n &\equiv \frac{\mu_j^n h}{2} \coth \frac{\mu_j^n h}{2\sigma_j^n}. \end{aligned} \quad (28)$$

We define the discrete variants of the initial condition (15) and boundary conditions (16) and we realise them as follows:

$$V_j^0 = \max(S_j - K, 0), \quad 1 \leq j \leq J-1 \quad (29)$$

and

$$\left. \begin{aligned} V_0^n &= g_0(t_n) \\ V_J^n &= g_1(t_n) \end{aligned} \right\} 0 \leq n \leq N. \quad (30)$$

The system (28), (29), (30) can be cast as a linear matrix system:

$$A^n U^{n+1} = F^n, \quad n \geq 0 \text{ with } U^0 \text{ given} \quad (31)$$

and we solve this system using LU decomposition, for example. A discussion of this topic with algorithms and implementation in C++ can be found in [3]. Summarising, the scheme (28) uses constant meshes in both space and time, centred differencing in space and backwards in time (fully implicit) marching. Furthermore, we use exponential fitting (see [2]) to ensure that the method remains stable and accurate for problems with small diffusion parameter or large convection parameter (This is the case of *convection dominance*). We note that equation (29) is the discrete payoff function for a call option. It plays the role of the discrete initial condition for the finite difference scheme (28), (29), (30).

Finally, we remark that scheme (28), (29), (30) is first-order accurate in space and time. For higher-order methods for one-factor and multi-factor Black Scholes PDEs, see [5] and [7].

6. SOFTWARE DESIGN AND IMPLEMENTATION ISSUES

What happens when we have set up the system of equations (28), (29), (30)? In general, we implement the schemes in some modern object-oriented programming language, for example C++ or C# for use in production environments although languages such as Matlab and Mathematica are used for building and testing prototypes. Many pricing libraries have been developed during the last twenty-five years in C++ and its popularity can be attributed to the fact that it is an ISO standard and it is very efficient. It is a big language and the learning curve is steep.

A discussion of the software activities involved when designing software systems in computational finance is outside the scope of this article. See [3] for some applications to PDEs and to the finite difference method.

7. CONCLUSIONS AND FUTURE SCENARIOS: COMPUTATIONAL FINANCE AND RESEARCH MATHEMATICS

We have written this article to show some of the mathematical, numerical and computational techniques that are used to price and hedge financial derivatives. We have focused on a small subset but important subset, namely the Black Scholes PDE and its numerical approximation using the finite difference method. There are many challenges and opportunities in this field in my opinion for applied and numerical mathematicians, computer scientists and engineers

in the coming years as we enter an era of distributed and parallel computing.

REFERENCES

- [1] Black, F. and M. Scholes 1973 The pricing of options and corporate liabilities. *Journal of Political Economy* 81, 637-659.
- [2] Duffy, D. J. 1980 Uniformly Convergent Difference Schemes for Problems with a Small Parameter in the Leading Derivative. PhD thesis. Trinity College Dublin.
- [3] Duffy, D. J. 2004 Financial Instrument Pricing using C++. John Wiley and Sons. Chichester.
- [4] Duffy, D. J. 2004A A critique of the Crank-Nicolson scheme, strengths and weaknesses for financial engineering. *Wilmott Magazine*. July 2004. pp. 68-76.
- [5] Duffy, D. J. 2006 Finite difference methods in financial engineering. John Wiley and Sons. Chichester.
- [6] Duffy, D.J. and Kienitz, J. 2009 Monte Carlo Frameworks, Building Customisable High Performance C++ Applications. John Wiley and Sons. Chichester.
- [7] Duffy, D.J. and Germani, A. 2013 C# in Financial Markets. John Wiley and Sons. Chichester.
- [8] Haug, E. 2007 The Complete Guide to Option Pricing Formulas. McGraw-Hill. New York.
- [9] Hull J. 2006 Options, Futures and other Derivatives. Sixth Edition. Pearson. Upper Saddle River, New Jersey.
- [10] Ilin, A.M, Kalashnikov, A.S. and Oleinik, O.A. 1962 Linear Equations of the Second Order of Parabolic Type, (translation) *Russian Mathematical Surveys* 17 (no. 3) 1-143.
- [11] Wilmott, P. 2006 Paul Wilmott on Quantitative Finance. John Wiley and Sons. Chichester.

Daniel J. Duffy is founder of Datasim Financial, Amsterdam. He has been using C++ since 1989 and has a PhD in Numerical Analysis from Trinity College. His interests are in computation, mathematics and their applications.

DATASIM EDUCATION BV, SCHIPLUIDENLAAN 4, 1062 HE AMSTERDAM,
THE NETHERLANDS

E-mail address: `dduffy@datasim.nl`