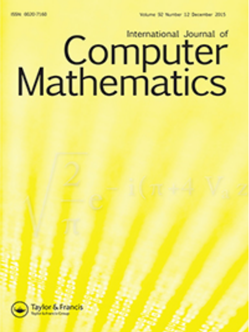

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: http://www.tandfonline.com/loi/gcom20

BENCHOP – The BENCHmarking project in option pricing

Lina von Sydow, Lars Josef Höök, Elisabeth Larsson, Erik Lindström, Slobodan Milovanović, Jonas Persson, Victor Shcherbakov, Yuri Shpolyanskiy, Samuel Sirén, Jari Toivanen, Johan Waldén, Magnus Wiktorsson, Jeremy Levesley, Juxi Li, Cornelis W. Oosterlee, Maria J. Ruijter, Alexander Toropov & Yangzhang Zhao

To cite this article: Lina von Sydow, Lars Josef Höök, Elisabeth Larsson, Erik Lindström, Slobodan Milovanović, Jonas Persson, Victor Shcherbakov, Yuri Shpolyanskiy, Samuel Sirén, Jari Toivanen, Johan Waldén, Magnus Wiktorsson, Jeremy Levesley, Juxi Li, Cornelis W. Oosterlee, Maria J. Ruijter, Alexander Toropov & Yangzhang Zhao (2015) BENCHOP – The BENCHmarking project in option pricing, International Journal of Computer Mathematics, 92:12, 2361-2379, DOI: 10.1080/00207160.2015.1072172

To link to this article: http://dx.doi.org/10.1080/00207160.2015.1072172

Accepted author version posted online: 24 Aug 2015.
Published online: 21 Sep 2015.

International Journal of Computer Mathematics, 2015
Vol. 92, No. 12, 2361–2379, http://dx.doi.org/10.1080/00207160.2015.1072172

BENCHOP – The BENCHmarking project in option pricing[†]

Lina von Sydow[a*], Lars Josef Höök[a], Elisabeth Larsson[a], Erik Lindström[b], Slobodan Milovanović[a], Jonas Persson[c], Victor Shcherbakov[a], Yuri Shpolyanskiy[d], Samuel Sirén[c], Jari Toivanen[e,f], Johan Waldén[g], Magnus Wiktorsson[b], Jeremy Levesley[h], Juxi Li[h], Cornelis W. Oosterlee[i,j], Maria J. Ruijter[i], Alexander Toropov[d] and Yangzhang Zhao[h]

[a]*Department of Information Technology, Uppsala University, Uppsala, Sweden;* [b]*Centre for Mathematical Sciences, Lund University, Lund, Sweden;* [c]*SUNGARD FRONT ARENA, Stockholm, Sweden www.sungard.com/frontarena;* [d]*ORC GROUP, Stockholm, Sweden, www.orc-group.com and ITMO University, St Petersburg, Russia;* [e]*Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland;* [f] *The Institute for Computational and Mathematical Engineering (ICME), Stanford University, Stanford, CA, USA;* [g]*Haas School of Business, University of California Berkeley, Berkeley, CA, USA;* [h]*Department of Mathematics, University of Leicester, Leicester, UK;* [i]*Centrum Wiskunde & Informatica, Amsterdam, The Netherlands;* [j]*Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands*

(*Received 30 January 2015; revised version received 18 June 2015; accepted 23 June 2015*)

The aim of the BENCHOP project is to provide the finance community with a common suite of benchmark problems for option pricing. We provide a detailed description of the six benchmark problems together with methods to compute reference solutions. We have implemented fifteen different numerical methods for these problems, and compare their relative performance. All implementations are available on line and can be used for future development and comparisons.

Keywords: option pricing; numerical methods; benchmark problem; Monte Carlo method; Fourier-method; finite difference method; radial basis function

2010 AMS Subject Classifications: 65-02; 91G60; 91G20

1. Introduction

The research on numerical methods for option pricing problems has been extensive over the last decades and there is now a plethora of methods targeting various types of options. However, there is a lack of cross comparisons between methods and a similar lack of common benchmarks to evaluate new approaches.

The aim of BENCHOP is to provide the finance community with a set of common benchmark problems that can be used both for comparisons between methods and for evaluation of new methods. Furthermore, in order to facilitate comparisons, MATLAB implementations of a wide range of existing methods for each benchmark problem will be made available through the BENCHOP web site www.it.uu.se/research/project/compfin/benchop.

*Corresponding author. Email: lina@it.uu.se
[†]The list of authors is organized in the following way: Project leader; Main contributors in alphabetical order; Code contributors in alphabetical order.

We also aim for BENCHOP to serve as a takeoff for future development of methods in option pricing. We expect future papers in the field to use the BENCHOP codes and problems to evaluate performance. In this way, we can contribute to a more uniform and comparable evaluation of the relative strengths and weaknesses of proposed methods.

The benchmark problems have been chosen in such a way as to be relevant both for practitioners and researchers. They should also be possible to implement with a reasonable effort. We have selected problems with respect to a number of features that may be numerically challenging. These are early exercise properties, barriers, discrete dividends, local volatility, stochastic volatility, jump diffusion, and two underlying assets. We have also included evaluation of hedging parameters in one of the problems, as this adds additional difficulties.

In this paper, we present the benchmark problems with sufficient detail so that other people can solve them in the future. We also provide analytical solutions where such are available or methods for computing accurate reference solutions otherwise. Each problem is solved using MATLAB implementations of a number of already existing numerical methods, and timing results are provided as well as error plots. For details of the methods, we refer to the original papers and additional notes at the BENCHOP web site. The codes are not fully optimized, and the numerical results should not be interpreted as competition scores. We rather see it as a synoptical exposition of the qualities of the different methods.

In Section 2, we state and motivate the benchmark problems while the numerical methods are briefly presented in Section 3. Section 4 is dedicated to the presentation of the numerical results and finally in Section 5, we discuss the results. In Appendix 1, we present how the reference values are computed for the different problems and in Appendix 2, we discuss how the local volatility surface is computed for one of the problems.

2. Benchmark problems

In this section, we state each of the six benchmark problems. In the mathematical formulations, we let S represent the actual (stochastic) asset price realization, whereas s is the asset price variable in the PDE formulation of the problem, t is the time (with $t = 0$ representing today), r is the risk free interest rate, σ is the volatility, W is a Wiener process, u is the option price as a function of s, K is the strike price, and T is the time of maturity. The payoff function $\phi(s)$ is the value of the option at time T. In Problem 4, V is the stochastic variance variable, and v is the variance value in the PDE-formulation.

In practice, the asset value today, S_0, is a known quantity, while the strike price K can take on different values. In the benchmark problem descriptions below we have chosen to fix all parameters, and then solve for different values of S_0 to simulate pricing of options that are ‘in the money’, ‘at the money’ and ‘out of the money’. The initial values are not given in the problem descriptions, but for each table and figure in the numerical results section, the values that were used are listed.

2.1 *Problem 1: The Black–Scholes–Merton model for one underlying asset*

The celebrated Black–Scholes–Merton [4,39] option pricing model, developed in the early 1970’s, is arguably the most successful quantitative model ever introduced in social sciences, even initiating the new field of Financial Engineering, which occupies thousands of researchers in financial institutions and universities across the world.

A key property of the model is that by building on so-called no-arbitrage arguments, it allows the price of plain vanilla call and put options to be calculated using variables that are either directly observable or can be easily estimated. The model is still widely used as a benchmark, although more advanced models have been developed over the years to take into account real-world features of asset price dynamics, such as jumps and stochastic volatility (see below).

The Black–Scholes–Merton model has the advantage that closed form solutions exist for prices, as well as for hedging parameters, for some types of options. It has therefore been extensively used to test numerical methods that are then applied to more advanced problems. The computation of the hedging parameters (Greeks) is included in this benchmark problem as they are of significant practical interest and can be expensive and/or difficult to compute for some numerical methods.

Mathematical formulation

$$\text{SDE-setting:} \quad \mathrm{d}S = rS\,\mathrm{d}t + \sigma S\,\mathrm{d}W. \tag{1}$$

$$\text{PDE-setting:} \quad \frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru = 0. \tag{2}$$

Deliverables

The pricing problem should be solved for three types of options; (a) a European call option, (b) an American put option, and (c) a barrier option. For the European option also the most common hedging parameters $\Delta = \partial u/\partial s$, $\Gamma = \partial^2 u/\partial s^2$ and $\mathcal{V} = \partial u/\partial \sigma$ should be computed.

Parameter and problem specifications

For this problem we have two sets of model parameters, representing less and more numerically challenging situations, respectively.

$$\text{Standard parameters:} \quad \sigma = 0.15, \quad r = 0.03, \quad T = 1.0, \quad \text{and} \quad K = 100. \tag{3}$$

$$\text{Challenging parameters:} \quad \sigma = 0.01, \quad r = 0.10, \quad T = 0.25, \quad \text{and} \quad K = 100. \tag{4}$$

The three types of options are characterized by their exercise properties and payoff functions.

$$\begin{aligned}
&\text{(a) European call :} && \phi(s) = \max(s-K, 0).\\
&\text{(b) American put :} && \phi(s) = \max(K-s, 0),\\
& && u(s,t) \geq \phi(s), \quad 0 \leq t \leq T.\\
&\text{(c) Barrier call up-and-out :} && \phi(s) = \begin{cases} \max(s-K,0), & 0 \leq s < B \\ 0, & s \geq B \end{cases}, \quad B = 1.25K.
\end{aligned}$$

2.2 *Problem 2: The Black–Scholes–Merton model with discrete dividends*

A shortcoming of the classical Black-Scholes formula is that it is only valid if the underlying stock does not pay dividends, invalidating the approach for many stocks in practice. In some special cases, for example, when dividend yields are constant and paid continuously over time, closed form solutions can be derived for dividend paying stocks too, see [39]. Usually, however, numerical methods are needed to calculate the option's value.

In practice, dividends are paid at discrete points in time, and the size of the dividend payments depends on the performance of the firm. For example, a firm whose performance has been poor may be capital constrained and therefore choose not to make a dividend payment, as may a company that needs its capital for a new investment opportunity. Fairly advanced stochastic modeling may therefore be needed in practice to capture the dividend dynamics of a company. In numerical tests, it is common to abstract away from these issues and simply assume that the firm makes discrete proportional dividend payments (i.e. has a constant dividend yield).

Mathematical formulation

$$\text{SDE-setting:}\quad \mathrm{d}S = rS\,\mathrm{d}t + \sigma S\,\mathrm{d}W - \delta(t-\tau)D\,S\,\mathrm{d}t. \tag{5}$$

$$\text{PDE-setting:}\quad \frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru = 0, \tag{6}$$

In the SDE case, the (single) dividend at time τ enters explicitly, whereas in the PDE case, it is implicitly taken into account by enforcing

$$u(s,\tau^-) = u(s(1-D),\tau^+). \tag{7}$$

Deliverables

Prices should be computed for a) a European call option and b) an American call option.

Parameter and problem specifications

The dividend is defined by $\tau = 0.4$ and $D = 0.03$. We use the standard parameters (3) except for the expiration time, set to $T = 0.5$, together with standard payoff function $\phi(s) = \max(s-K,0)$ and European and American exercise properties respectively, see Problem 1.

2.3 *Problem 3: The Black–Scholes–Merton model with local volatility*

As mentioned earlier, the Black–Scholes–Merton model with a constant volatility does not reproduce market prices very well in practice. One discrepancy is the so-called volatility smile (which after the October 1987 crash is known to have turned into a smirk). If the implied volatility—the volatility in the Black–Scholes–Merton model that is consistent with the observed option price—is calculated for several options with the same exercise date but different strike prices, all options should under the classical assumptions of Black–Scholes–Merton have the same implied volatility. Instead, when plotted against the different strike prices, the curve is usually that of a U-shaped smile (or an L-shaped smirk).

As discussed in [10], an approach to address this discrepancy between model and data is to assume local volatility, that is, to allow the volatility of the underlying asset to depend instantaneously on the stock price s, and time t, generating a whole volatility surface. It is shown in [10] how to reverse engineer such a volatility surface from observed option prices.

Given a volatility surface, the general Black–Scholes–Merton no-arbitrage approach can be used to derive the option price, although closed form solutions will typically no longer exist. We provide two volatility surfaces with different properties in order to see how the numerical methods handle such variable volatility coefficients.

Mathematical formulation

The equations for the option price are identical to (1) and (2) except that here $\sigma = \sigma(s,t)$.

Deliverables

The price for a European call option should be computed in each case.

Parameter and problem specifications

The first local volatility surface is given by an explicit function

$$\sigma_I(s,t) = 0.15 + 0.15(0.5+2t)\frac{(s/100-1.2)^2}{(s/100)^2+1.44}. \tag{8}$$

The second local volatility surface $\sigma_{II}(s,t)$ is based on market data and does not have an explicit form. The local surface is computed from a parametrization of the implied volatility data. The exact steps in the computation and the specific parametrization are given in Appendix 2.

In both cases, we use $K = 100$ and $r = 0.03$, but the expiration times are different, with $T = 1$ for σ_I, and $T = 0.5$ for σ_{II}. The payoff function for a European call option is as before $\phi(s) = \max(s-K,0)$.

2.4 *Problem 4: The Heston model for one underlying asset*

The local volatility model allows for perfect matching of prices of European-style options but, just like the Black–Scholes–Merton model, also has its weaknesses. It does not perform very well for path dependent options and, moreover, there is clear evidence that in practice the volatility of asset prices is in itself random, beyond what can be simply be described as a function of time and underlying strike price [9,32,45]. The Heston model [20] assumes that in addition to the risk-factor that drives the value of the underlying asset, there is a another risk-factor that determines the underlying's instantaneous variance, V. The PDE formulation of the model is therefore two-dimensional. Note that in contrast to the previous models, the market in Heston's model is incomplete, and therefore additional assumptions about the market price of volatility risk are needed to determine the option price. The specific assumptions in [20] leads to the model below.

Mathematical formulation

SDE-setting:

$$\begin{aligned} \mathrm{d}S &= rS\,\mathrm{d}t + \sqrt{V}S\,\mathrm{d}W_1, \\ \mathrm{d}V &= \kappa(\theta - V)\,\mathrm{d}t + \sigma\sqrt{V}\,\mathrm{d}W_2, \end{aligned} \tag{9}$$

where W_1 and W_2 have correlation ρ.

PDE-setting:

$$\frac{\partial u}{\partial t} + \frac{1}{2}vs^2\frac{\partial^2 u}{\partial s^2} + \rho\sigma vs\frac{\partial^2 u}{\partial s\partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 u}{\partial v^2} + rs\frac{\partial u}{\partial s} + \kappa(\theta - v)\frac{\partial u}{\partial v} - ru = 0. \tag{10}$$

Deliverables

The price for a European call option should be computed.

Parameter and problem specifications

The model parameters are here given by $r = 0.03$, $\kappa = 2$, $\theta = 0.0225$, $\sigma = 0.25$, $\rho = -0.5$, $K = 100$, and $T = 1$. The payoff function for the European call option is $\phi(s, v) = \max(s - K, 0)$. With these parameters, the Feller condition is satisfied.

2.5 *Problem 5: The Merton jump diffusion model for one underlying asset*

The Merton model [40] addresses another difference between real world asset price dynamics and the (local volatility) Black–Scholes–Merton model. What was identified early on, is that stock prices occasionally experience dramatic movements over very short time periods, that is, they sometimes 'jump'. Such jumps make return distributions heavier-tailed than for pure diffusion processes, also in line with empirical observations and, as in the Heston model, causes the market to be incomplete, necessitating additional assumptions to price the option. The assumption used in [40] is that the underlying stock price follows a jump-diffusion process, where there is no risk-premium associated with jump risk. Under these conditions, the option price can be computed from a Partial-Integro Differential Equation (PIDE).

Mathematical formulation

SDE-setting:

$$\mathrm{d}S = (r - \lambda\xi)S\,\mathrm{d}t + \sigma S\,\mathrm{d}W + S\,\mathrm{d}Q, \tag{11}$$

where Q is a compound Poisson process with intensity $\lambda > 0$ and jump ratios that are log-normally distributed as $p(y) = 1/\sqrt{2\pi}y\delta\mathrm{e}^{-(\log y-\gamma)^2/2\delta^2}$ [50].

PIDE-setting:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + (r - \lambda\xi)s\frac{\partial u}{\partial s} - (r + \lambda)u + \lambda \int_0^\infty u(sy, \tau)p(y)\,\mathrm{d}y = 0. \tag{12}$$

Deliverables

The price for a European call option should be computed.

Parameter and problem specifications

The parameters to use are $r = 0.03$, $\lambda = 0.4$, $\gamma = -0.5$, $\delta = 0.4$, $\xi = \mathrm{e}^{\gamma+\delta^2/2} - 1$, $\sigma = 0.15$, $K = 100$, and $T = 1$. The payoff function is $\phi(s) = \max(s - K, 0)$.

2.6 *Problem 6: The Black–Scholes–Merton model for two underlying assets*

As an example of an option with more than one underlying, we use a spread option, which for $K = 0$ is called a Margrabe option [38]. This classic rainbow option has a payoff function that depends on two underlying assets, so the option price dynamic therefore depends on two risk-factors (as long as the two stocks' returns are not perfectly correlated). In contrast to the Heston and Merton models, the model is still within the class of complete market models, and the option price is therefore completely determined without further assumptions. The reason that the market is complete in this case, in contrast to the other multi risk-factor models we have introduced, is that two underlying risky assets may be used in the formation of a hedging portfolio, whereas only one such asset is available with the Heston and Merton Models.

Mathematical formulation

SDE-setting:

$$\begin{aligned} \mathrm{d}S_1 &= rS_1\,\mathrm{d}t + \sigma_1 S_1\,\mathrm{d}W_1, \\ \mathrm{d}S_2 &= rS_2\,\mathrm{d}t + \sigma_2 S_2\,\mathrm{d}W_2, \end{aligned} \tag{13}$$

where W_1 and W_2 have correlation ρ.

PDE-setting:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma_1^2 s_1^2 \frac{\partial^2 u}{\partial s_1^2} + \rho\sigma_1\sigma_2 s_1 s_2 \frac{\partial^2 u}{\partial s_1 \partial s_2} + \frac{1}{2}\sigma_2^2 s_2^2 \frac{\partial^2 u}{\partial s_2^2} + rs_1\frac{\partial u}{\partial s_1} + rs_2\frac{\partial u}{\partial s_2} - ru = 0. \tag{14}$$

Deliverables

The price for a European spread call option should be computed.

Parameter and problem specifications

The model parameters to use are $r = 0.03$, $\sigma_1 = \sigma_2 = 0.15$, $\rho = 0.5$, $K = 0$, and $T = 1$. The payoff function for the European call spread option is $\phi(s_1, s_2) = \max(s_1 - s_2 - K, 0)$.

3. Numerical methods

In Table 1, we display all methods that we have used. We also provide references to the original papers describing the methods. More information about the particular implementations used here can be found at www.it.uu.se/research/project/compfin/benchop.

4. Numerical results

For each benchmark problem, we have decided on three (or five) evaluation points s_i (or (s_i, s_j)). Each method must be tuned such that it delivers a solution $u(s_i)$ with a *relative error* less than

Table 1. List of methods used with abbreviations, marker symbol used in figures, and references.

Abbr.	Symbol	Method	References
Monte Carlo methods			
MC	- -*- -	Monte Carlo with Euler-Maruyama in time	[16]
MC-S	- -△- -	Monte Carlo with analytical solution/Euler-Maruyama/quadratic scheme in time and stratified sampling	[2,16,17,36,42,50]
QMC-S	- -×- -	Quasi Monte Carlo with analytical solution/Euler-Maruyama/quadratic scheme in time, stratified sampling, and precomputed quasi random numbers	[2,16,17,21,36,42,44,50]
Fourier methods			
FFT	-·□-·	Fourier method with FFTs	[6,31,33]
FGL	-·×-·	Fourier method with Gauss-Laguerre quadrature	[1 (Page 890),7,18,31], [34 (Section 2.1–2.2), 35]
COS	-·○-·	Fourier method based on Fourier cosine series and the characteristic function.	[11,12,52,53]
Finite difference methods			
FD	——	Finite differences on uniform grids with Rannacher smoothed CN in time	[23,51,59,64, Chapter 78]
FD-NU	- - - - -	Finite differences on quadratically refined grids with Rannacher smoothed CN / IMEX-CNAB in time	[22,49,51,55,56]
FD-AD	-·-·-·-	Adaptive finite differences with discontinuous Galerkin / BDF-2 in time	[19,22,26,46,47,62]
Radial basis function methods			
RBF	—×—	Global radial basis functions with non-uniform nodes and BDF-2 in time	[19,48]
RBF-FD	—◇—	Radial basis functions generated finite differences with BDF-2 in time	[13,14,19,41,58,63,65]
RBF-PUM	—△—	Radial basis functions partition of unity method with BDF-2 in time	[19,54,57]
RBF-LSML	—*—	Least-squares multi-level radial basis functions with BDF-2 in time	[19,27,28]
RBF-AD	—○—	Adaptive RBFs with CN in time	[8,24,43]
RBF-MLT	—□—	Multi-level radial basis functions treating time as a spatial dimension	[25,30,60]

10^{-4} in these points. We have not put any restrictions on the error in the rest of the domain. Due to this freedom, some codes have been tuned to narrowly target these points, while others (sometimes automatically) are tuned to achieve an evenly distributed error. Then the codes are run (on the same computer system) and the execution times are recorded. Each code is run four times, and the execution time reported in Tables 2–5 is the average of the last three runs. This is because the first time a MATLAB script is executed in a session it takes a bit longer.

In the tables, we also show the approximate number of correct digits p in the result. This quantity is computed as $p = [-\log_{10} e_r]$, where e_r is the maximum relative error and $[\cdot]$ indicates rounding. The maximum relative error is computed as

$$e_r = \max_i \left| \frac{u(s_i) - u^{\mathrm{ref}}(s_i)}{u^{\mathrm{ref}}(s_i)} \right|.$$

For the Monte Carlo methods where the error is not deterministic, the errors are averaged over the different runs. In some cases, a method was not able to reach a relative error of 10^{-4} within reasonable time (1 hour), but a lower target 10^{-3} was attainable. These results are marked with a * in Tables 2–5. The execution time we report in the tables for Monte Carlo methods is the

Table 2. Problem 1. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90$, 100, 110 for the standard parameters and at $s = 97$, 98, 99 for the challenging parameters.

	Standard parameters			Challenging parameters		
Method	(a) European	(b) American	(c) Up-and-out	(a) European	(b) American	(c) Up-and-out
MC	*5.7e + 02 (3)	–	–	×	–	–
MC-S	1.5e + 01 (4)	×	×	1.6e + 01 (4)	9.8e − 02 (16)	×
QMC-S	5.7e − 01 (5)	×	–	6.3e − 01 (6)	1.2e + 02 (16)	–
FFT	1.3e − 03 (6)	–	–	1.3e − 03 (7)	–	–
FGL	3.5e − 03 (14)	7.6e − 01 (5)	–	1.8e − 02 (13)	2.2e + 00 (16)	–
COS	1.8e − 04 (5)	2.7e − 02 (4)	1.8e − 02 (4)	2.6e − 04 (4)	2.3e − 01 (5)	2.5e − 04 (4)
FD	1.8e − 02 (4)	7.6e − 02 (4)	2.5e − 02 (4)	5.1e + 01 (4)	1.1e − 02 (11)	5.1e + 01 (4)
FD-NU	9.2e − 03 (4)	5.8e − 02 (4)	1.6e − 02 (4)	5.0e − 02 (5)	6.0e − 03 (11)	5.2e − 02 (5)
FD-AD	9.7e − 03 (4)	4.3e − 02 (4)	9.6e − 03 (4)	1.0e + 01 (4)	9.1e − 03 (4)	2.0e + 00 (4)
RBF	6.2e − 02 (4)	4.6e + 00 (4)	1.4e − 01 (4)	7.7e + 01 (4)	–	6.6e + 01 (4)
RBF-FD	2.9e − 01 (4)	1.3e + 00 (4)	2.8e − 01 (4)	3.3e + 01 (5)	9.6e − 01 (4)	5.1e + 00 (4)
RBF-PUM	2.8e − 02 (4)	3.6e + 00 (4)	5.4e − 02 (4)	3.4e + 00 (5)	4.5e + 00 (4)	1.9e + 00 (4)
RBF-LSML	4.2e − 02 (4)	–	3.0e − 02 (4)	7.5e + 00 (5)	–	–
RBF-AD	7.9e − 01 (4)	1.7e + 01 (5)	2.4e + 01 (5)	3.3e + 00 (4)	1.2e + 00 (7)	1.6e + 01 (4)
RBF-MLT	1.6e + 01 (4)	–	2.4e + 02 (4)	*3.3e + 02 (4)	–	*1.9e + 03 (3)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

Table 3. Problem 1. Computational time to compute hedging parameters $\Delta = \frac{\partial u}{\partial s}$, $\Gamma = \frac{\partial^2 u}{\partial s^2}$ and $\mathcal{V} = \frac{\partial u}{\partial \sigma}$ that have a relative error $< 10^{-4}$ at $t = 0$ and $s = 90$, 100, 110 for the standard parameters and at $s = 97$, 98, 99 for the challenging parameters.

	Standard parameters			Challenging parameters		
Method	$\Delta = \frac{\partial u}{\partial s}$	$\Gamma = \frac{\partial^2 u}{\partial s^2}$	$\mathcal{V} = \frac{\partial u}{\partial \sigma}$	$\Delta = \frac{\partial u}{\partial s}$	$\Gamma = \frac{\partial^2 u}{\partial s^2}$	$\mathcal{V} = \frac{\partial u}{\partial \sigma}$
MC-S	3.0e + 00 (5)	*5.7e + 01 (4)	1.7e + 01 (4)	3.3e + 00 (5)	×	*1.8e + 01 (3)
QMC-S	5.7e − 01 (5)	*1.0e + 00 (4)	6.3e − 01 (5)	6.3e − 01 (6)	×	*6.7e − 01 (4)
FFT	1.3e − 03 (6)	1.2e − 03 (5)	2.1e − 03 (5)	1.2e − 03 (7)	1.2e − 03 (5)	2.5e − 03 (6)
FGL	3.2e − 03 (14)	2.9e − 03 (14)	2.9e − 03 (14)	1.8e − 02 (14)	1.8e − 02 (11)	2.2e − 02 (11)
COS	1.8e − 04 (4)	2.4e − 04 (5)	2.6e − 04 (5)	2.8e − 04 (4)	4.4e − 04 (5)	4.6e − 04 (5)
FD	1.9e − 02 (5)	1.4e − 02 (5)	2.8e − 02 (4)	4.9e + 01 (5)	2.5e + 02 (4)	5.0e + 02 (5)
FD-NU	7.8e − 03 (4)	8.9e − 03 (4)	1.8e − 02 (4)	6.5e − 02 (4)	5.0e − 01 (4)	1.6e + 00 (4)
FD-AD	1.0e − 02 (4)	9.9e − 03 (4)	2.4e − 02 (4)	1.0e + 01 (4)	4.9e + 01 (4)	9.0e + 01 (4)
RBF	6.6e − 02 (4)	6.9e − 02 (4)	7.9e − 02 (4)	9.3e + 01 (4)	3.1e + 02 (5)	7.0e + 02 (4)
RBF-FD	3.0e − 01 (4)	2.5e − 01 (4)	5.6e − 01 (4)	3.4e + 01 (4)	3.7e + 01 (4)	1.0e + 02 (4)
RBF-PUM	2.7e − 02 (4)	3.2e − 02 (4)	1.0e − 01 (4)	3.1e + 00 (5)	6.2e + 00 (4)	1.1e + 01 (4)
RBF-LSML	1.3e − 01 (4)	2.7e − 01 (4)	7.1e − 02 (5)	–	–	–
RBF-AD	3.4e + 00 (4)	3.4e + 01 (4)	5.0e + 01 (4)	3.6e + 00 (5)	1.8e + 01 (4)	2.1e + 01 (4)
RBF-MLT	1.8e + 01 (4)	1.8e + 01 (4)	*3.2e + 01 (4)	4.2e + 02 (4)	*3.4e + 02 (3)	×

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

time to compute the result for one evaluation point, whereas for other methods it is the time to compute the result for all evaluation points.

In order to see how the errors behave away from the evaluation points, solutions are also plotted for a range of values in Figures 1–8. The figures show the *absolute errors* evaluated at the integer values between $s = 60$ and $s = 160$. No figure is shown for the second local volatility case in Problem 3, because there the local volatility result is only valid for a particular S_0, not

Table 4. Problems 2 and 3. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90, 100, 110$.

	Discrete dividends				Local volatility			
Method	European call		American call		Smooth		Implied	
MC-S	*6.0e+01	(3)	–		×		×	
QMC-S	1.6e+00	(4)	–		–		–	
FFT	1.5e−03	(7)	–		–		–	
FGL	3.6e−03	(14)	8.1e−02	(6)	–		–	
COS	8.8e−04	(5)	2.4e−03	(4)	1.7e−02	(4)	–	
FD	2.0e−02	(4)	1.5e−02	(4)	2.2e−02	(4)	1.2e+00	(4)
FD-NU	1.6e−02	(4)	1.5e−02	(4)	3.7e−02	(4)	8.9e−01	(4)
FD-AD	2.1e−02	(4)	2.3e−02	(5)	3.6e−02	(4)	8.8e−01	(4)
RBF	2.3e−01	(4)	1.1e−01	(4)	5.5e−02	(4)	2.4e+00	(4)
RBF-FD	4.2e−01	(4)	2.7e+00	(4)	2.2e+01	(4)	1.1e+02	(4)
RBF-PUM	3.3e−02	(4)	3.0e−02	(4)	1.4e−01	(4)	9.0e−01	(4)
RBF-LSML	5.0e−01	(4)	1.2e+00	(4)	1.7e−01	(4)	4.0e+00	(4)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

Table 5. Problems 4, 5, and 6. Computational time to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $s = 90, 100, 110$ for the Heston and Merton models, and to compute a solution u that has a relative error $< 10^{-4}$ at $t = 0$ and $(s_1, s_2) = (100, 90), (100, 100), (100, 110), (90, 100), (110, 100)$ for the spread option.

Method	Heston		Merton		Spread	
MC-S	×		1.6e+01	(4)	*2.9e+01	(4)
QMC-S	–		3.4e−01	(4)	1.8e+00	(5)
FFT	3.3e−03	(5)	2.1e−03	(5)	–	
FGL	3.8e−03	(14)	3.1e−03	(13)	3.1e−03	(14)
COS	3.4e−04	(4)	2.2e−04	(4)	1.5e−03	(4)
FD	–		–		–	
FD-NU	4.3e+00	(4)	1.4e−01	(4)	7.4e+01	(4)
FD-AD	–		–		4.7e+01	(4)
RBF	1.9e+01	(4)	–		7.7e+01	(4)
RBF-FD	–		–		2.2e+03	(4)
RBF-PUM	4.3e+00	(4)	–		1.3e+01	(5)

Note: The numbers within parentheses indicate the approximate number of correct digits in the result. A ‘ – ’ indicates not implemented, while ‘ × ’ means implemented, but not accurate.

over a range. The vertical axis range in the figures is adjusted to the values that are plotted, but the lower limit is not allowed to be lower than 10^{-20}. Errors falling below that value are not visible in the figures.

The experiments have been performed on the Tintin cluster at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX), Uppsala University. The cluster consists of 160 dual AMD Opteron 6220 (Bulldozer) nodes. All codes are implemented (serially) in MATLAB. The names of the respective codes for each problem are indicated by the boldfaced heading over each (group of) plot(s). This generic name is then combined with an acronym for the particular method as for example `BSeuCallUI_RBF.m`.

5. Discussion

Monte Carlo methods. MC methods are easy to implement in any number of dimensions, but the slow convergence rate, $\mathcal{O}(1/\sqrt{N})$ for standard MC, makes it computationally expensive to reach

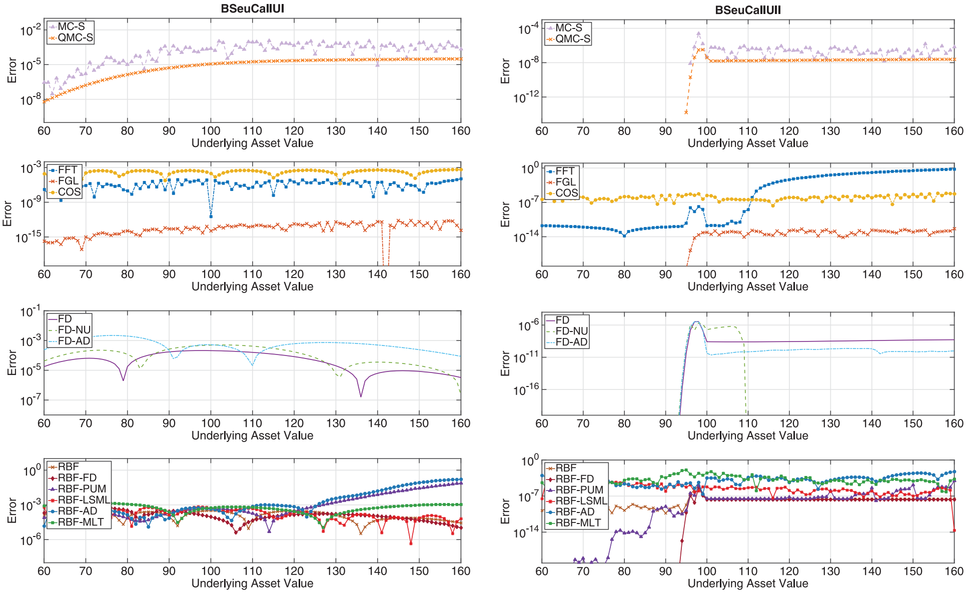

Figure 1. Problem 1(a) European call option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

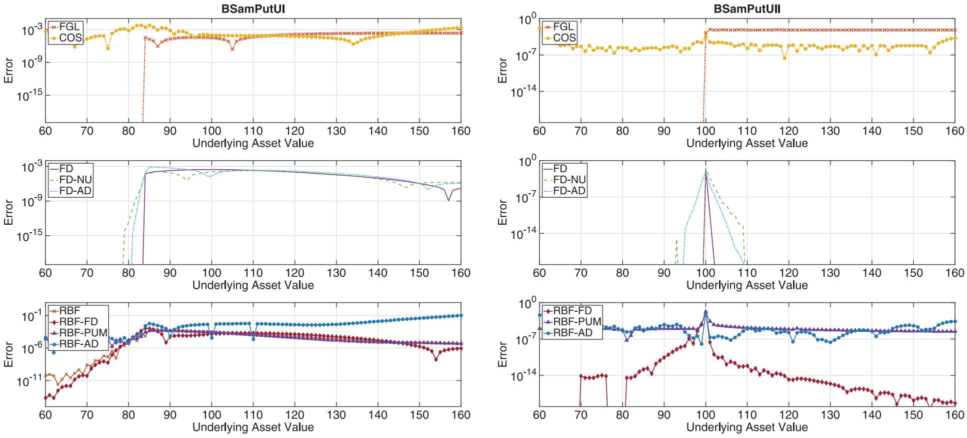

Figure 2. Problem 1(b) American put option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

the requested tolerance of 10^{-4}. The most challenging problems for the MC methods were the path dependent options, the hedging parameter Γ, and the local volatility.

Because MC methods scale linearly with the number of dimensions, they are increasingly competitive in higher dimensions. Furthermore, there are a lot of specialized techniques that can be applied to improve performance. An example of this is the QMC-S method that is comparable to some of the other methods already in one dimension, and the fastest method apart from the Fourier methods for the two-dimensional spread option.

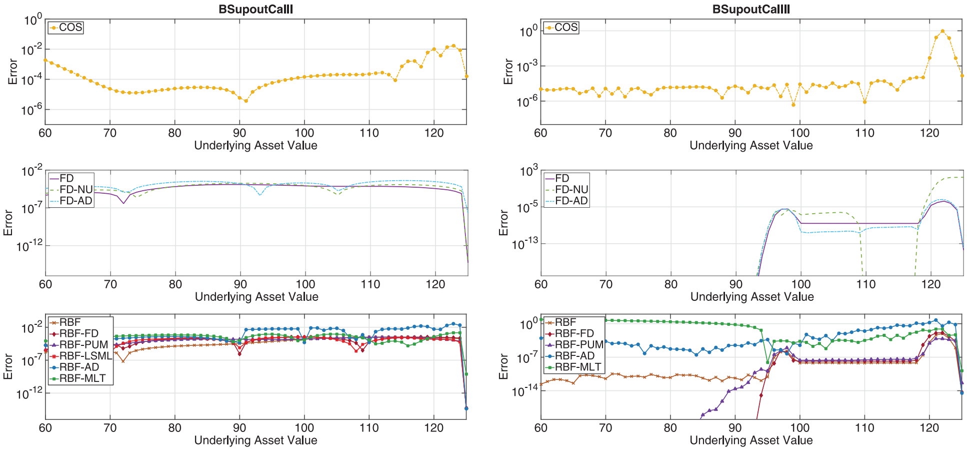

Figure 3. Problem 1(c) Up-and-out call option. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute error in the solution u for $t = 0$ and $60 \le s \le 125$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

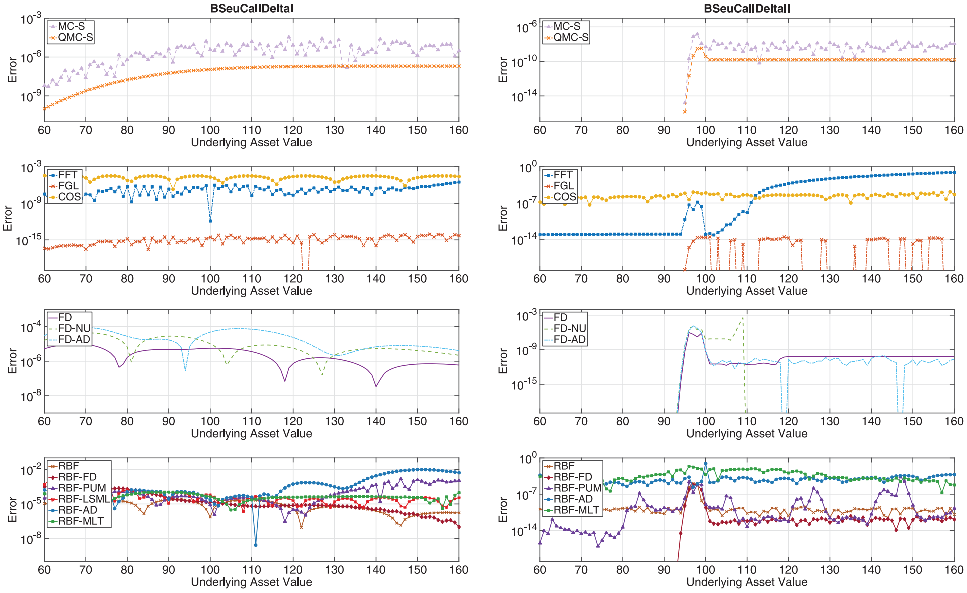

Figure 4. Problem 1(a) European call option. Errors in the hedging parameter Δ. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \le s \le 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

Fourier methods. Fourier methods (FM) rely on the availability of the characteristic function (ChF) of the underlying stochastic process. These are available for all problems except Problem 3, local volatility. However, in the recent publication [53], the stochastic process is approximated by a second order weak Taylor scheme, for which there exists an analytic solution for the ChF. This method was applied to the smooth local volatility function, but could not easily be used for the implied local volatility. Apart from Problem 3, the problems that were most challenging for the FM were the American and Up-and-out options.

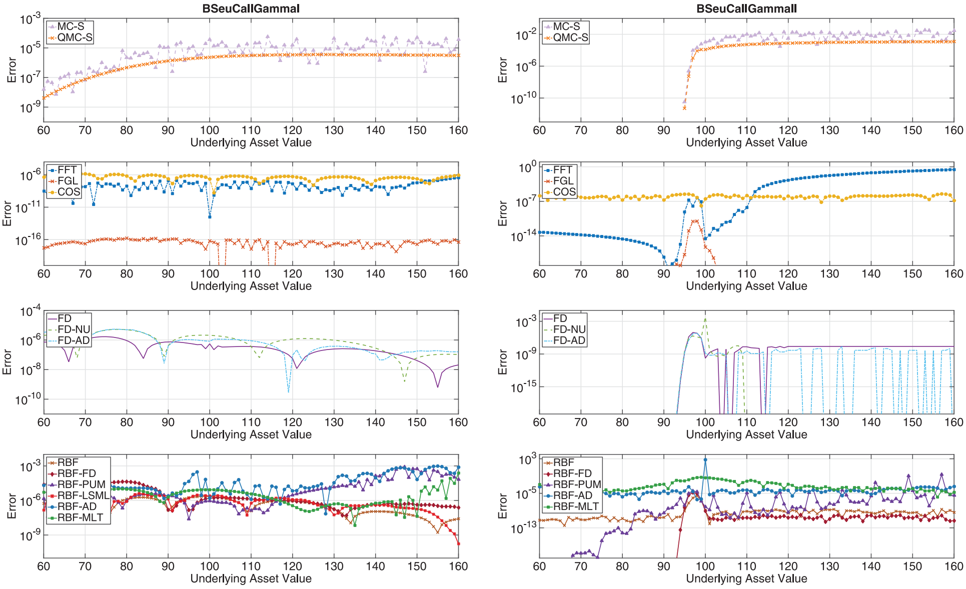

Figure 5. Problem 1(a) European call option. Errors in the hedging parameter Γ. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \leq s \leq 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

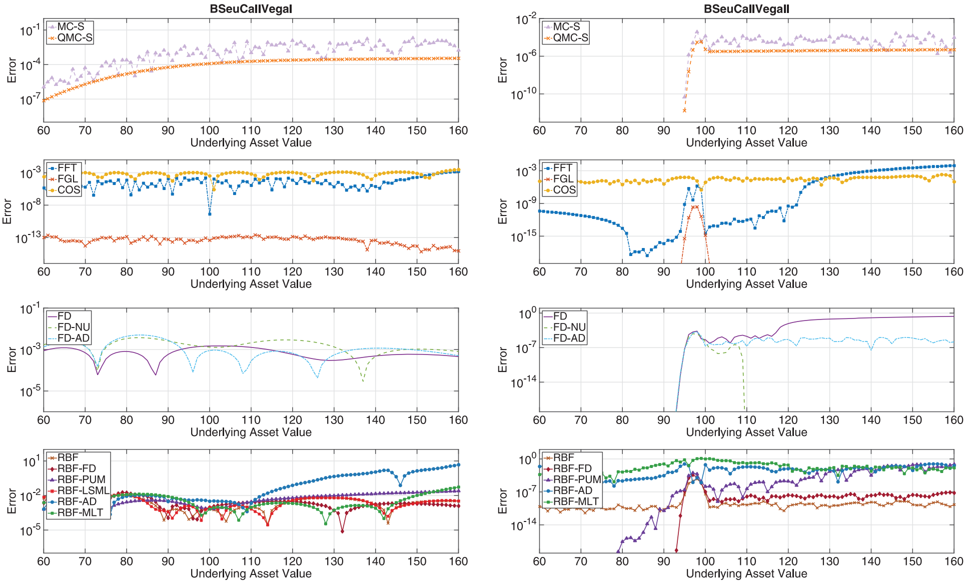

Figure 6. Problem 1(a) European call option. Errors in the hedging parameter $\mathcal{V}$. For each problem, results for standard parameters are shown to the left, and for challenging parameters to the right. Absolute errors in the hedging parameters for $t = 0$ and $60 \leq s \leq 160$ when the relative error is less than 10^{-4} at $t = 0$ and $s = 90$, 100, 110 for standard parameters and $s = 97$, 98, 99 for challenging parameters.

The FM are all very fast and especially the FGL-method is also highly accurate. The fastest FM, the COS method, is the overall fastest method in all cases but two. The competitiveness of the FM is even more pronounced for the two-dimensional problems, the Heston model and the spread option.

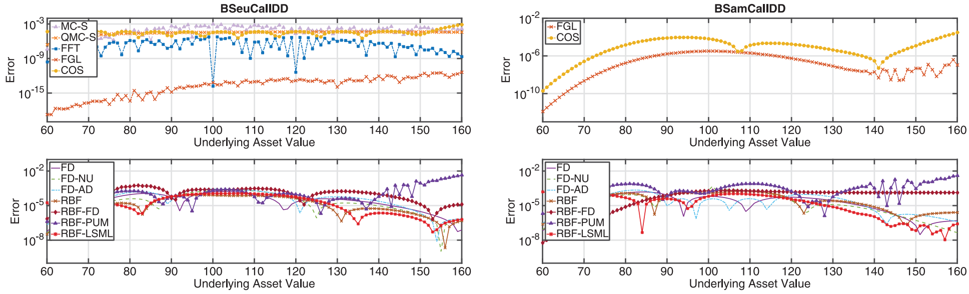

Figure 7. Problem 2(a) European call with dividends (left) and 2(b) American call with dividends (right). Error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$.

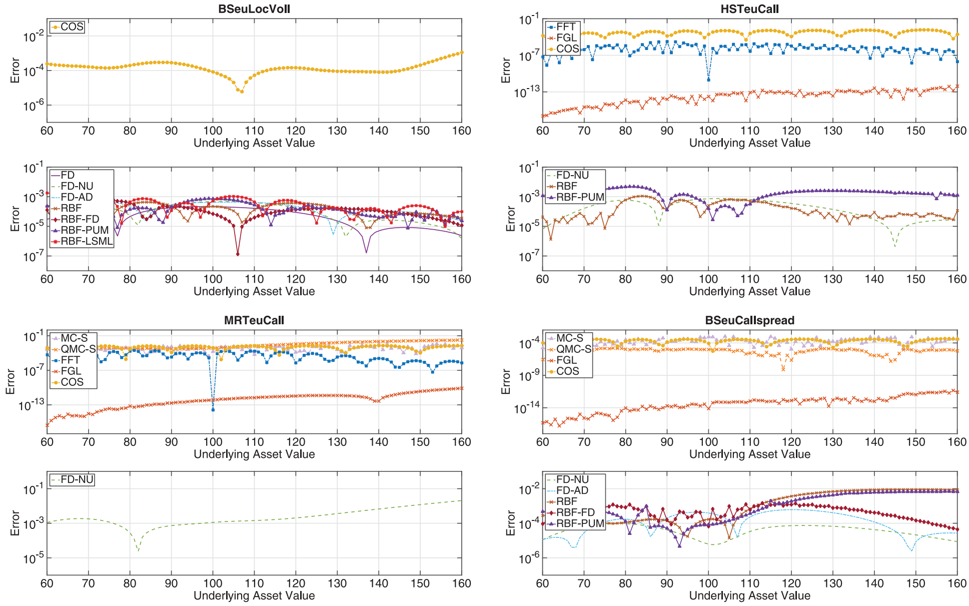

Figure 8. Problem 3 Local volatility, smooth function (top left), Problem 4 Heston (top right), Problem 5 Merton (bottom left), and Problem 6 Spread option (bottom right). Absolute error in the solution u for $t = 0$ and $60 \le s \le 160$ when the relative error in u is less than 10^{-4} at $t = 0$ and $s = 90, 100, 110$ for Local volatility and Merton, and for Heston with variance $v = 0.0225$. For the spread option, the error is measured in $(s_1, s_2) = (100, 90)$, $(100, 100)$, $(100, 110)$, $(90, 100)$, $(110, 100)$, and is plotted for $60 \le s_1 \le 160$, and $s_2 = 100$.

Finite difference methods. FD methods rely on the use of structured (possibly non-uniform) grids and are straightforward to implement. The FD-NU is the only BENCHOP method that solved all the problems. The computational times are low in all cases, and for two problems FD-NU or FD-AD is the overall fastest method. For the FD methods in general the challenging parameter set in Problem 1 is the most difficult feature to handle.

For Problem 1 the usage of a nonuniform grid, (FD-NU and FD-AD), is superior to using a uniform grid (FD) but for Problems 2 and 3, using a uniform grid is sometimes faster. The FD method has not been implemented for the two-dimensional problems, but we believe that FD-NU and FD-AD would be faster than FD in these cases thanks to the possibility of local refinement.

Radial basis function methods. RBF methods are flexible with respect to node locations and choice of basis function. This makes it possible to tune the methods to achieve well for particular targets, but it can also be hard to make a good choice. Problem 5, Merton jump diffusion has not been implemented in any RBF method, but this could be done. The problems that are challenging for RBF methods have non-smooth solutions or very sharp gradients, such as American options and the challenging parameter set in Problem 1.

The RBF methods as a group are slower than the FD methods for the one-dimensional problems. However, the results of the fastest RBF method, RBF-PUM, is in the favorable cases of the same order as those of the FD methods. In two dimensions the RBF-PUM method is as fast as or faster than the implemented FD methods. Potentially, the RBF-PUM method will be even more competitive in higher dimensions.

Acknowledgments

We would like to thank Institut Mittag-Leffler for supporting the workshop we organized on ‘Mathematical and Numerical modeling in Finance’ (http://www.mittag-leffler.se/?q = 0609), where the idea for the BENCHOP project was conceived and where the work was initiated.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project snic2014-3-73.

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing*, Dover publications, New York, 1972.
[2] L.B.G. Andersen, *Efficient simulation of the Heston stochastic volatility model*, Tech. rep., 2007.
[3] T. Björk, *Arbitrage Theory in Continuous Time*, 3rd ed., Oxford University Press, New York, 2009.
[4] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Econ. 81 (1973), pp. 637–654.
[5] P. Carr, R. Jarrow, and R. Myneni, *Alternative characterizations of American put options*, Math. Finance 2 (1992), pp. 87–106.
[6] P. Carr and D. Madan, *Option valuation using the fast Fourier transform*, J. Comput. Finance 2 (1999), pp. 61–73.
[7] P. Carr, H. Geman, D. Madan, L. Wu, and M. Yor, *Option Pricing Using Integral Transforms*, 2003, a slideshow presentation available at: http://www.math.nyu.edu/research/carrp/papers/pdf/integtransform.pdf.
[8] R. Chan, *Pricing options under jump-diffusion models by adaptive radial basis functions*, Tech. rep., University of Bath, Department of Economics, 2010, Bath Economics Research Working papers; 06/10.
[9] R. Cont, *Empirical properties of asset returns: stylized facts and statistical issues*, Quant. Finance 1 (2001), pp. 223–236.
[10] B. Dupire, *Pricing with a smile*, Risk 7 (1994), pp. 18–20.
[11] F. Fang and C.W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput. 31 (2008), pp. 826–848.
[12] F. Fang and C.W. Oosterlee, *Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions*, Numer. Math. 114 (2009), pp. 27–62.
[13] N. Flyer, E. Lehto, and S. Blaise, *A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere*, J. Comput. Phys. 231 (2012), pp. 4078–4095. Available at http://www.sciencedirect.com/science/article/pii/S0021999112000587.
[14] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys. 230 (2011), pp. 2270–2285. Available at http://www.sciencedirect.com/science/article/pii/S0021999110006789.
[15] J. Gatheral and A. Jacquier, *Arbitrage-free SVI volatility surfaces*, Quant. Finance 14 (2014), pp. 59–71.
[16] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Applications of mathematics: stochastic modelling and applied probability, Springer, 2004. Available at http://books.google.se/books?id = e9GWUsQkPNMC.
[17] P. Glasserman and J. Staum, *Conditioning on one-step survival for barrier option simulations*, Oper. Res. 49 (2001), pp. 923–937. Available at http://dx.doi.org/10.1287/opre.49.6.923.10018.
[18] G.H. Golub and J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp. 23 (1969), pp. 221–230 + s1–s10.
[19] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Solving Ordinary Differential Equations, Springer, 2008. Available at https://books.google.se/books?id = cfZDAAAAQBAJ.

[20] S.L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financial Stud. 6 (1993), pp. 327–343.

[21] L.J. Höök, T. Johnson, and T. Hellsten, *Randomized quasi-Monte Carlo simulation of fast-ion thermalization*, Comput. Sci. Disc. 5 (2012), pp. 014010. Available at http://stacks.iop.org/1749-4699/5/i = 1/a = 014010.

[22] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett. 17 (2004), pp. 809–814. Available at http://dx.doi.org/10.1016/j.aml.2004.06.010.

[23] S. Ikonen and J. Toivanen, *Pricing American options using LU decomposition*, Appl. Math. Sci. 1 (2007), pp. 2529–2551.

[24] A. Iske and J. Behrens, *Grid-free adaptive semi-Lagrangian advection using radial basis functions*, Comput. Math. Appl. 43 (2002), pp. 319–327.

[25] A. Iske and J. Levesley, *Multilevel scattered data approximation by adaptive domain decomposition*, 39 (2005), pp. 187–198.

[26] E. Larsson, *Option pricing using the discontinuous Galerkin method for time-integration*, Tech. rep., Department of Information Technology, Uppsala University, 2013.

[27] E. Larsson and S.M. Gomes, *A Least Squares Multi-Level Radial Basis Function Method with Applications in Finance*, 2015, manuscript in preparation.

[28] E. Larsson, S. Gomes, A. Heryudono, and A. Safdari-Vaighani, *Radial basis function methods in computational finance*, in *Proc. CMMSE 2013*, Almería, Spain, 12pp., 2013.

[29] M. Lauko and D. Ševčovič, *Comparison of numerical and analytical approximations of the early exercise boundary of American put options*, ANZIAM J. 51 (2010), pp. 430–448. Available at http://dx.doi.org/10.1017/S1446181110000854.

[30] Q.T. Le Gia, I.H. Sloan, and H. Wendland, *Multiscale approximation for functions in arbitrary Sobolev spaces by scaled radial basis functions on the unit sphere*, Appl. Comput. Harmon. Anal. 32 (2012), pp. 401–412.

[31] R.W. Lee, *Option pricing by transform methods: Extensions, unification, and error control*, J. Comput. Finance 7 (2004), pp. 51–86, an augmented version is available at http://www.math.uchicago.edu/˜rl/dft.pdf.

[32] E. Lindström, H. Madsen, and J. Nielsen, *Statistics for Finance*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2015.

[33] E. Lindström and H. Wu, *A Fourier method for valuation of options under parameter uncertainty*, submitted for publication .

[34] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst, *Sequential calibration of options*, Tech. Rep. 2006:21, Centre for Mathemtical Sciences, Lund University, 2006.

[35] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst, *Sequential calibration of options*, Comput. Statist. Data Anal. 52 (2008), pp. 2877–2891.

[36] F.A. Longstaff and E.S. Schwartz, *Valuing American options by simulation: A simple least-squares approach*, Rev. Financ. Stud. 14 (2001), pp. 113–147. Available at http://rfs.oxfordjournals.org/content/14/1/113.abstract.

[37] R. Lord and C. Kahl, *Optimal fourier inversion in semi-analytical option pricing*, J. Comput. Finance 10 (2007), pp. 1–30.

[38] W. Margrabe, *The value of an option to exchange one asset for another*, J. Finance 33 (1978), pp. 177–186.

[39] R.C. Merton, *Theory of rational option pricing*, Bell J. Econom. Man. Sci. 4 (1973), pp. 141–183.

[40] R.C. Merton, *Option pricing when underlying stock returns are discontinuous*, J. Financial Econ. 3 (1976), pp. 125–144.

[41] S. Milovanović and L. von Sydow, *Radial basis function generated finite differences for option pricing problems*, 2015. Manuscript in preparation.

[42] K.-S. Moon, *Efficient Monte Carlo algorithm for pricing barrier options*, Commun. Korean Math. Soc. 23 (2008), pp. 285–294.

[43] S.L. Naqvi, *Adaptive radial basis function interpolation for time-dependent partial differential equation*, Ph.D. thesis, University of Leicester, 2013.

[44] H. Niederreiter, *Constructions of (t,m,s)–nets and (t,s)–sequences*, Finite Fields Appl. 11 (2005), pp. 578–600. Available at http://www.sciencedirect.com/science/article/pii/S1071579705000043.

[45] P. Nystrup, H. Madsen, and E. Lindström, *Stylized Facts of Financial Time Series and Hidden Markov Models in Continuous Time*, Quant. Finance, 2015.

[46] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci. 10 (2007), pp. 173–183.

[47] J. Persson and L. von Sydow, *Pricing American options using a space-time adaptive finite difference method*, Math. Comput. Simulation 80 (2010), pp. 1922–1935.

[48] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, J. Comput. Appl. Math. 222 (2008), pp. 82–93.

[49] O. Pironneau, *On the transport-diffusion algorithm and its applications to the Navier-Stokes equations*, Numer. Math. 38 (1981/82), pp. 309–332. Available at http://dx.doi.org/10.1007/BF01396435.

[50] E. Platen and N. Bruti-Liberati, *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, Stochastic Modelling and Applied Probability, Vol. 64, Springer-Verlag, Berlin, 2010.

[51] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math. 43 (1984), pp. 309–327.

[52] M.J. Ruijter and C.W. Oosterlee, *Two-dimensional Fourier cosine series expansion method for pricing financial options*, SIAM J. Sci. Comput. 34 (2012), pp. B642–B671.

[53] M.J. Ruijter and C.W. Oosterlee, *Numerical Fourier method and second-order Taylor scheme for backward SDEs in finance*, Working paper available at SSRN, submitted for publication, 2014.

[54] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, *A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications*, J. Sci. Comp. (2014), pp. 1–27.

[55] S. Salmi and J. Toivanen, *IMEX schemes for pricing options under jump-diffusion models*, Appl. Numer. Math. 84 (2014), pp. 33–45. Available at http://dx.doi.org/10.1016/j.apnum.2014.05.007.

[56] S. Salmi, J. Toivanen, and L. von Sydow, *An IMEX-scheme for pricing options under stochastic volatility models with jumps*, SIAM J. Sci. Comput. 36 (2014), pp. B817–B834. Available at http://dx.doi.org/10.1137/130924905.

[57] V. Shcherbakov and E. Larsson, *Radial basis function partition of unity methods for pricing vanilla basket options*, Tech. Rep. 2015-001, Division of Scientific Computing, Department of Information Technology, Uppsala University, 2015.

[58] A.I. Tolstykh and D.A. Shirobokov, *On using radial basis functions in a 'finite difference mode' with applications to elasticity problems*, Comput. Mech. 33 (2003), pp. 68–79.

[59] A. Toropov, D. Ivanov, and Y. Shpolyanskiy, *Application of theoretical options pricing to algorithmic trading strategies*, Sci. Tech. J. Infor. Tech. Mech. Opt. (2013), pp. 136–141.

[60] A. Townsend and H. Wendland, *Multiscale analysis in Sobolev spaces on bounded domains with zero boundary values*, IMA J. Numer. Anal. 33 (2013), pp. 1095–1114.

[61] R. Villiger, *Valuation of American call options*, Wilmott Magazine 3 (2006), pp. 64–67.

[62] L. von Sydow, *On discontinuous Galerkin for time integration in option pricing problems with adaptive finite differences in space*, in *Numerical Analysis and Applied Mathematics: ICNAAM 2013*, *AIP Conference Proceedings*, vol. 1558, American Institute of Physics (AIP), Melville, NY, 2013, pp. 2373–2376.

[63] X. Wang, *Finite differences based on radial basis functions to price options*, Master's thesis, Uppsala University, 2014.

[64] P. Wilmott, *On Quantitative Finance*, 2nd ed., John Wiley & Sons Ltd., Chichester, 2006.

[65] G.G.B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys. 212 (2006), pp. 99–123. Available at http://www.sciencedirect.com/science/article/pii/S0021999105003116.

Appendix 1. The methods used for computing the reference values used in the comparisons

For many of the benchmark problems here, there are analytical or semi-analytical solutions. For these cases, we state the closed form expressions. For the cases lacking analytical solutions, we describe the numerical method that was used for accurate enough computation of a reference solution. MATLAB codes for each problem are available at the BENCHOP web page.

A.1 *Problem 1: The Black–Scholes–Merton model for one underlying asset*

For the *European call* option, the closed form expression for the option price is given in [4]

$$\Pi^c_{BS}(t,S,K,T,r,\sigma^2) = SN\left(d^+\left(\frac{S}{K},T-t\right)\right) - Ke^{-r(T-t)}N\left(d^-\left(\frac{S}{K},T-t\right)\right), \tag{A1}$$

where

$$d^+(x,y) = \frac{1}{\sigma\sqrt{y}}\left(\log(x) + \left(r+\frac{\sigma^2}{2}\right)y\right), \tag{A2}$$

$$d^-(x,y) = \frac{1}{\sigma\sqrt{y}}\left(\log(x) + \left(r-\frac{\sigma^2}{2}\right)y\right), \tag{A3}$$

and where $N(x)$ is the cumulative distribution function for the standard normal distribution. The hedging parameters can be found through differentiation, leading to

$$\Delta_{BS} = \frac{\partial \Pi^c_{BS}}{\partial S} = N\left(d^+\left(\frac{S}{K},T-t\right)\right), \tag{A4}$$

$$\Gamma^c_{BS} = \frac{\partial^2 \Pi^c_{BS}}{\partial S^2} = \frac{\phi(d^+(S/K,T-t))}{S\sqrt{(T-t)\sigma^2}}, \tag{A5}$$

$$\mathcal{V}^c_{BS} = \frac{\partial \Pi^c_{BS}}{\partial \sigma} = S\phi\left(d^+\left(\frac{S}{K},T-t\right)\right)\sqrt{T-t}, \tag{A6}$$

where $\phi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ is the density of the standard normal distribution.

For the *American put* option, there is no closed form solution. Different analytical and semi-analytical approximations to the location of the early exercise boundary are analyzed and compared in [29]. Here, we use a relation from [5, Theorem 1.1], where it is shown that the American put price can be decomposed into a European put price and the early exercise premium. The general European put price is defined by

$$\Pi^{p}_{BS}(t,S,K,T,r,\sigma^2) = -SN\left(-d^{+}\left(\frac{S}{K},T-t\right)\right) + K\mathrm{e}^{-r(T-t)}N\left(-d^{-}\left(\frac{S}{K},T-t\right)\right), \tag{A7}$$

The American put price becomes

$$\Pi^{p}_{BS-A}(t,S,K,T,r,\sigma^2) = \Pi^{p}_{BS}(t,b_p(t),K,T,r,\sigma^2) + rK\int_t^T \mathrm{e}^{-r(u-t)}N\left(-d^{-}\left(\frac{b_p(t)}{b_p(u)},u-t\right)\right)\,\mathrm{d}u, \tag{A8}$$

where $b_p(t)$, $0 \le t < T$ is the (unknown) optimal exercise level at time t, and $b_p(T) = K$. The location of the exercise boundary is determined by solving the following non-linear integral equation numerically:

$$\Pi^{p}_{BS-A}(t,S,K,T,r,\sigma^2) = K - b_p(t). \tag{A9}$$

This equation is solved by an implicit trapezoidal method, where the implicit step due to the monotonicity in $b_p(t)$ can be found by binary search. This leads to a robust albeit slow method for obtaining the optimal exercise level b_p on a fine time grid from $t = T$ to $t = 0$. Then, for arbitrary initial stock values satisfying $S(0) > b_p(0)$, (A8) provides the option value. For $S(0) \le b_p(0)$ the value is set to the exercise value $K - S(0)$.

For the *barrier call up–and–out* option, a closed form expression for the option price can be found in [3, Theorem 18.12 p. 271].

$$\begin{aligned}\Pi^{c}_{BS-UO}(t,S,K,T,r,\sigma^2) &= \Pi^{c}_{BS}(t,S,K,T,r,\sigma^2) - \Pi^{c}_{BS}(t,S,B,T,r,\sigma^2)\\ &\quad - \left(\frac{B}{S}\right)^{2r/\sigma^2-1}\left(\Pi^{c}_{BS}\left(t,\frac{B^2}{S},K,T,r,\sigma^2\right) - \Pi^{c}_{BS}\left(t,\frac{B^2}{S},B,T,r,\sigma^2\right)\right).\end{aligned} \tag{A10}$$

A.2 *Problem 2: The Black–Scholes–Merton model with discrete dividends*

For the *European call* option with one proportional discrete dividend payment, there is a closed form solution [3, Proposition 16.6 p. 235]. Assuming that the dividend is paid out at time τ, where $t < \tau < T$, we have

$$\Pi^{c}_{BS-EDD}(t,S,K,T,r,\sigma^2) = \Pi^{c}_{BS}(t,S(1-D),K,T,r,\sigma^2). \tag{A11}$$

Note that the option price is independent of when during the contract period the dividend occurs.

The *American Call* option with one dividend payment at $\tau = \alpha T$, where $0 < \alpha < 1$, can be valuated semi analytically [61].

$$\begin{aligned}\Pi^{c}_{BS-ADD}(0,S,K,T,r,\sigma,D,\alpha) &= (1-D)SN_2\left(-d^{+}\left(\frac{S}{S^*_\tau},\alpha T\right), d^{+}\left(\frac{(1-D)S}{K},T\right), -\sqrt{\alpha}\right)\\ &\quad - K\mathrm{e}^{-rT}N_2\left(-d^{-}\left(\frac{S}{S^*_\tau},\alpha T\right), d^{-}\left(\frac{(1-D)S}{K},T\right), -\sqrt{\alpha}\right)\\ &\quad + \Pi^{c}_{BS}((1-\alpha)T,S,S^*_\tau,T,r,\sigma^2),\end{aligned} \tag{A12}$$

where $N_2(x,y,\rho)$ is the cumulative distribution function for the bi-variate normal distribution with zero mean and covariance matrix

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

The above formula depends on the unknown variable S^*_t, which can be estimated from the following nonlinear problem:

$$S^*_\tau - K = \Pi^{c}_{BS}(T-\tau,(1-D)S^*_\tau,K,T,r,\sigma^2).$$

A.3 *Problem 3: The Black–Scholes–Merton model with local volatility*

For a general local volatility function, there is no closed form solution. The reference values have in this case been computed to high accuracy by a selection of the contributed deterministic methods with different numerical approximations in space, different treatments of the boundary, and different approximations in time. In this way, we can be reasonably certain that numerical bias has been eliminated.

A.4 *Problem 4: The Heston model for one underlying asset*

The (almost exact) Heston price is computed through inverse Fourier transform using the Gauss-Laguerre quadrature method (FGL) with 1000 quadrature points in combination with an optimized choice of integration path in the complex plane stock value for stock value. The integration path is chosen so that is goes through the uniquely given saddle point of the integrand (see [34,37]).

A.5 *Problem 5: The Merton jump diffusion model for one underlying asset*

From [40] we have that the price under Merton jump diffusion is given by a weighted sum of modified Black-Scholes prices

$$\Pi^c_{ME}(t,S,K,T,r,\lambda,\gamma,\delta^2,\sigma^2)=\sum_{n=0}^{\infty}\mathrm{e}^{-\lambda(T-t)}\frac{(\lambda(T-t))^n}{n!}\Pi^c_{BSE}\left(t,(\xi+1)^n\mathrm{e}^{-\lambda(T-t)\xi}S,K,T,\sigma^2+\delta^2\frac{n}{T-t}\right),\quad\text{(A13)}$$

where $\xi=\mathrm{e}^{\gamma+\delta^2/2}-1$. By using a few hundred terms, a highly accurate price approximation can be computed.

A.6 *Problem 6: The Black–Scholes–Merton model for two underlying assets*

For the particular case of $K=0$, a closed form solution for the European call spread option is given in [38] as

$$\Pi^c_{BS-SO}(t,S_1,S_2,T,\sigma_1,\sigma_2,\rho)=\Pi^c_{BS}(t,S_1,S_2,T,0,\sigma_1^2-2\rho\sigma_1\sigma_2+\sigma_2^2).\quad\text{(A14)}$$

Appendix 2. The second local volatility function used in Problem 3

The second local volatility function is based on a stochastic volatility inspired (SVI) parametrization [15] of the implied volatility surface. This approach to local volatility is widely used by practitioners, because it is relatively easy to calibrate to data, and there are techniques to eliminate different modes of arbitrage.

A.7 *The given SVI parametrization*

The global total implied variance surface in terms of the time of maturity T and the log moneyness $x=\log(K/F_T)$ in the forward price $F_T=S_0\mathrm{e}^{rT}$ is given by

$$w_g(T,x)=a+\frac{r-\ell}{2}(x-m)+\frac{r+\ell}{2}\sqrt{(x-m)^2+p^2},\quad\text{(A15)}$$

where a, r, ℓ, m, p are parameters that depend on T. Here, the calibrated parameters are given by

$$a=0.01+0.03\sqrt{T+0.04},\quad r=0.06(1-0.87\sqrt{T}),$$
$$\ell=0.31(1-0.7\sqrt{T}),\quad m=0.03+0.01T,$$
$$p=0.15(0.4+0.6\sqrt{T+0.04}).$$

A.8 *Constructing the local volatility surface*

In order to compute the local surface, we apply a transformation that corresponds to Dupire's formula [10] for the SVI parametrization.

$$w_{\mathrm{local}}(x,T)=\frac{w_g(T,x)+T\dfrac{\partial w_g}{\partial T}(T,x)}{\left(1-\dfrac{x\dfrac{\partial w_g}{\partial x}}{2w_g}\right)^2-\left(\dfrac{\dfrac{\partial w_g}{\partial x}}{2w_g}\right)^2\left(\left(\dfrac{w_gT}{2}+1\right)^2-1\right)+\dfrac{T\frac{\partial^2 w_g}{\partial x^2}}{2}}.\quad\text{(A16)}$$

This gives us a local volatility surface in terms of T and x.

A.9 *Using the local volatility surface*

When we use the local volatility surface, we replace K by s and T by t. In order to see what that implies for the log moneyness, we need to go back to the definition. There we had $x = \log(K/F_T)$, then when we use the local volatility surface

$$x(s, S_0, t) = \log \frac{s}{F_t(S_0)} = \log \frac{s}{S_0 \mathrm{e}^{rt}}.$$

The local volatility is now given by

$$\sigma(s, t) = \sqrt{w_{\text{local}}(x(s, S_0, t), t)}. \tag{A17}$$

Note that $\sigma(s, t)$ cannot be directly evaluated for very small values of s. A function that evaluates this volatility surface can be downloaded from the BENCHOP web site.