6

Numerical Computation of Model Prices

6.1 Overview

In this chapter we develop a new pricing algorithm to compute model prices for the derivatives contracts previously discussed. Here, we distinguish, as before, between contracts with unconditional and conditional exercise rights. The distinction is made because of the separate fundamental calculation procedure for these prices. Whereas derivatives with unconditional exercise rights can be calculated in terms of the general characteristic function $\psi(\mathbf{x}_t, z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)$ and in terms of the relevant moment-generating function[128], respectively, without evaluating any integral at all if the characteristic function is known in closed form, we need for option-type contracts to apply a numerical integration scheme in order to calculate their model prices. Carr and Madan (1999) showed in their prominent article a very convenient method to compute option prices for a given strike range, using the FFT. The advantage in applying the FFT to option-pricing problems, is its considerable computational speed improvement compared to other numerical integration schemes. Due to the payoff transform methodology, we use another pricing algorithm, which shares the same desirable, numerical properties of the FFT. Unfortunately, implementing the pricing approach according to Lewis (2001), it is necessary to impose the transform with respect to the strike. Therefore, one cannot use the FFT any longer to obtain option prices in one pass for a strike range[129].

[128] See Section 5.2.

[129] See Lee (2004), p. 61. However, comparing the structure in equation (4.21) it is possible to obtain model prices with the help of a FFT procedure for different levels of $g(\mathbf{x}_t)$.

In order to circumvent this problem within the payoff-transform pricing approach, we need an another numerical algorithm. Therefore, we incorporate in our pricing algorithm the IFFT, to compute model prices for different strike values[130]. Furthermore, to enhance the quality of results[131], the fractional Fourier Transform of Bailey and Swarztrauber (1994) is used. This refinement was introduced by Chourdakis (2005) in pricing equity option prices with the transformed option price methodology of Carr and Madan (1999).

However, we sometimes encounter the problem that $\psi(\mathbf{x}_t, z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)$ cannot be calculated in closed form[132]. For these cases, we implement a Runge-Kutta solver in our IFFT pricing algorithm. This algorithm is then used to compute the relevant values for different z in $\psi(\mathbf{x}_t, z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)$ by solving the ODEs (2.40) and (2.41) numerically and providing the procedure with the needed values.

6.2 Contracts with Unconditional Exercise Rights

As explained in Section 5.2.2 all contracts with unconditional exercise rights can be calculated as mere function evaluations of the general characteristic function $\psi(\mathbf{x}_t, z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)$, its first order derivative with respect to z, and for integro-linear payoff functions with the help of the first order derivative $\psi_z\left(\mathbf{x}_t, z, w_0^A(z), \mathbf{w}^A(z), 0, \mathbf{0}_M, \tau\right)$. As shown, these unconditional expectations can be obtained by contour integration in closed form. Thus, we do not need to develop a numerical integration routine at all in order to calculate the relevant model prices. The calculations reduce in these cases to

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(\mathbf{x}_s)\,\mathrm{d}s}\right] = \psi(\mathbf{x}_t, 0, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau),$$

[130] We find it natural to use the FFT and the IFFT algorithm to obtain the desired Fourier Transformation. Other numerical integration schemes are also possible, like for example the numerical integration via Laguerre polynomials as used in Tahani (2004).

[131] The ordinary IFFT pricing algorithm suffers, like the particular FFT algorithm, from the fixed scale of increments of strike values and transformation variable, which is discussed in Section 6.3.1.

[132] This could be the case e.g. for some subordinated processes r_t or for jump components where $\mathbb{E}_{\mathbf{J}}\left[\psi^*(z, w_0, \mathbf{w}, g_0, \mathbf{g}, \mathbf{J}, \tau)\right]$ cannot be solved explicitly.

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(\mathbf{x}_s)\,\mathrm{d}s} g\left(\mathbf{x}_T\right)\right] = \frac{\psi_z(\mathbf{x}_t, 0, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)}{\imath},$$

and

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(\mathbf{x}_s)\,\mathrm{d}s} \gamma(T)\right] = \frac{\psi_z(\mathbf{x}_t, 0, w_0^A(0), \mathbf{w}^A(0), 0, \mathbf{0}_M, \tau)}{\imath},$$

for arbitrary times to maturity τ. For normal contracts, the discount rate used in the characteristic function is based on the short rate $r\left(\mathbf{x}_t\right)$ and is zero for futures-style contracts. In case of an average-rate contract where the underlying is the geometric average of the short rate, we have to use the characteristic function with a modified discount rate $r^A\left(\mathbf{x}_t\right)$.

If the general characteristic function cannot be expressed in closed form although defined by a system of ODEs, we apply a numerical algorithm to evaluate the needed values. In this case we implement a Runge-Kutta solver for the system of ODEs (2.40) and (2.41).

6.3 Contracts with Conditional Exercise Rights

6.3.1 Calculating Option Prices with the IFFT

We start with the integral representation of the general option valuation formula (4.21). Since we are interested in calculating option prices in one pass for a given strike range simultaneously with the IFFT, we have to reduce the presence of K in the integral to the expression $e^{\imath z\alpha(K)}$ for both exponential-linear, linear, and integro-linear type payoff functions. In the case of coupon-bond options and swaptions we have to divide the payoff function up into A separate parts. The alternative representation of the valuation formula is

$$V(\mathbf{x}_t, t, T) = \frac{e^{\alpha(K)d}}{\pi} \int_0^\infty e^{\imath z\alpha(K)} \hat{g}(z) \psi(\mathbf{x}_t, -z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)\,\mathrm{d}z, \tag{6.1}$$

with

$$\mathcal{F}^{g(\mathbf{x}_T)}\left[G\left(\mathbf{x}_T\right)\right] = e^{(d+\imath z)\alpha(K)} \hat{g}(z),$$

and $\alpha(K) = K$ for the case of a floating-rate based contract and an asian-type contract, respectively, and $\alpha(K) = \ln[K]$ for a yield-based contract[133]. The

[133] See equation (5.21).

parameter d is chosen in a way to eliminate all dependency of $\alpha(K)$ in $\hat{g}(z)$, which is crucial for the IFFT algorithm to work properly[134]. A first problem might arise using multi-valued functions, e.g. the complex-valued logarithm, square-root, and the confluent hypergeometric function $\mathrm{KU}(a;b;y)$. Thus, we have to carefully keep track of the integration path to avoid any discontinuities[135]. However, using a numerical algorithm to compute the particular values of the characteristic function such as a Runge-Kutta algorithm we do not encounter these problems[136].

The first step in deriving the IFFT pricing algorithm is to truncate the integration domain as

$$f(\alpha(K)) \approx \int_0^{\omega} e^{\imath z\alpha(K)} \hat{g}(z) \psi(\mathbf{x}_t, -z, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)\, \mathrm{d}z. \tag{6.2}$$

Applying an U-point approximation with increment $\Delta = \frac{\omega}{U}$, we discretize the domain of the transform variable into

$$z_u = \left(u - \frac{1}{2}\right) \Delta + \imath z_i$$

with $u = 1, \ldots, U$ and z_i corresponding to a fixed value for which the Fourier-transformed payoff function exists. The integration interval $[0, \infty]$ is then replaced with a discrete, truncated region such that the integrand of $f(\alpha(K))$ is negligible for z_U. Hence, the discrete approximation to equation (6.2) is

$$\begin{aligned} f(\alpha(K)) &\approx \sum_{u=1}^{U} e^{\imath z_u \alpha(K)} \hat{g}(z_u) \psi(\mathbf{x}_t, -z_u, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau)\, \Delta \\ &= \Delta e^{-z_i \alpha(K)} \sum_{u=1}^{U} e^{\imath(u-1)\, \Delta\alpha(K)} e^{\frac{\imath\Delta}{2}\, \alpha(K)} \hat{g}_u \psi_u, \end{aligned} \tag{6.3}$$

[134] Otherwise, the IFFT algorithm is not applicable to the valuation problem at hand. Fortunately, we are able to reduce the dependency of K in the particular integrals to the specific term $e^{\imath z\alpha(K)}$, for all contracts discussed in Chapter 3.

[135] This topic is covered comprehensively in Nagel (2001), Appendix 4.

[136] In case of the Fong and Vasicek (1991a) model, we made the same experience as mentioned in Tahani (2004), Footnote 4, and compute values of the characteristic function with help of an explicit Runge-Kutta algorithm in the first place. Thus, besides the prevention of discontinuities, the Runge-Kutta algorithm can be more efficient than the explicit computation of the confluent hypergeometric function.

with

$$\hat{g}_u = \hat{g}(z_u) \qquad \text{and} \qquad \psi_u = \psi(\mathbf{x}_t, -z_u, w_0, \mathbf{w}, g_0, \mathbf{g}, \tau).$$

The sum above is commonly referred to as a discrete inverse Fourier Transformation[137] of the function $e^{\frac{\imath\Delta}{2}\alpha(K)}\hat{g}_u\psi_u$. We also want to mention that in computing this sum we eventually obtain the option price for only one particular strike value K. Since we are interested in calculating option prices for a strike range we also have to discretize $\alpha(K)$, which yields

$$\alpha_v = \alpha(K_1) + (v-1)\eta,$$

with step size η and $v = 1, \ldots, U$[138]. Thus, inserting the explicit expression for α_v inside the brackets of equation (6.3) gives

$$\begin{aligned} f(\alpha_v) &= \Delta\, e^{-z_i\alpha_v} \sum_{u=1}^{U} e^{\imath(u-1)\,\Delta(\alpha_1+(v-1)\eta)} e^{\frac{\imath\Delta}{2}(\alpha_1+(v-1)\eta)} \hat{g}_u\psi_u \\ &= \Delta\, e^{-z_i\alpha_v} e^{\frac{\imath\Delta\eta}{2}(v-1)} \sum_{u=1}^{U} e^{\imath(u-1)(v-1)\,\Delta\eta} e^{\imath\,\Delta\alpha_1(u-\frac{1}{2})} \hat{g}_u\psi_u. \end{aligned} \tag{6.4}$$

The form of $f(\alpha_v)$ is almost ready to be inserted into the IFFT algorithm.

The IFFT algorithm is developed to calculate simultaneously the discrete inverse Fourier Transformation for a range of values α_v. The main advantage is that it reduces the number of calculations from an order of U^2 to the order of $U\log_2[U]$, which makes a significant difference in computational speed[139]. It efficiently computes the sum

$$f(v, \mathbf{h}) = \frac{1}{U}\sum_{u=1}^{U} e^{\imath(u-1)(v-1)\frac{2\pi}{U}} h_u \qquad \text{for } v = 1, \ldots, U. \tag{6.5}$$

[137] Although we defined the transform operations in Section 2.4 vice versa, in this chapter we rely on the term discrete *inverse* transform, which belongs to engineering disciplines and is in line with the expression used afterwards for the IFFT.

[138] We use the same discretization scheme for $\alpha(K)$ as used in Lee (2004). The advantage, in contrast to the discretization schemes applied in Carr and Madan (1999) and Raible (2000), is the possibility to adjust the numerical scheme for the lower bound of the strike rates. Thus, one does not necessarily have to compute option prices for negligible strike rates, which is a more efficient procedure.

[139] See Cooley and Tukey (1965).

Introducing the vectors

$$\mathbf{u} = \mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ U \end{pmatrix},$$

equation (6.5) can be displayed in a more compact form, which is

$$\mathbf{f}(\mathbf{h}) = \text{IFFT}[\mathbf{h}], \tag{6.6}$$

with $\mathbf{h} \in \mathbb{C}^U$.

By comparing equation (6.5) with (6.4), we obviously need the relation

$$\Delta\eta = \frac{2\pi}{U},$$

in order to apply the IFFT algorithm properly to equation (6.4). Because $\frac{2\pi}{U}$ remains constant for a fixed number of points U, we have only the freedom to choose either Δ or η independently. Thus, there is a tradeoff between the accuracy of the calculated results and the coarseness of the strike-value grid. According to these considerations, more accurate results of option prices corresponding to specific strike rates have to be paid with more points in the integration scheme due to the rule $U \times 2^n$. This rule ensures that the algorithm computes option prices for specific strike values and illustrates the exponential cost for more accurate results. Calculating the same number of option prices, most of them outside a desired strike range, entails a substantial waste of computational time[140].

To give a more compact writing, we use henceforth the vectors $\boldsymbol{\alpha} = (\alpha_v)_{v=1}^U$, $\hat{\mathbf{g}} = (\hat{g}_u)_{u=1}^U$ and $\boldsymbol{\psi} = (\psi_u)_{u=1}^U$. Eventually, the vector $\mathbf{V}(\mathbf{x}_t, t, T)$ containing the option values for different strikes, can be computed as

$$\begin{aligned} \mathbf{V}(\mathbf{x}_t, t, T) = & \frac{U\,\Delta\, e^{(d-z_i)\boldsymbol{\alpha}}}{\pi} \\ & \odot \text{ Re}\left[e^{\frac{\pi\imath}{U}(\mathbf{v}-1)} \odot \text{IFFT}[e^{\imath\,\Delta\alpha_1(\mathbf{u}-\frac{1}{2})} \odot \hat{\mathbf{g}} \odot \boldsymbol{\psi}] \right], \end{aligned} \tag{6.7}$$

where the operator $\odot$ denotes the vector-dot product of two arbitrary vectors of the same length. This pricing algorithm is already capable of calculating

[140] This particular problem is addressed in the next section.

option prices. However, as stated before, equation (6.7) displays the problem of computing option prices for many irrelevant strike rates, given a desired level of accuracy.

6.3.2 Refinement of the IFFT Pricing Algorithm

The purpose of this subsection is to solve the problem of the inverse relationship of Δ and η mentioned in the last section. The numerical efficiency can be enhanced by using a modified version of the ordinary IFFT algorithm to ensure that all calculated option prices are at least within an interval of relevant strike values. Bailey and Swarztrauber (1994) developed a method based on the FFT to choose Δ and η independently. Their method, called the fractional Fourier Transformation, henceforth denoted as the FRFT, incorporates a new auxiliary parameter ζ[141], which successfully dissects the otherwise fixed relation $\Delta\eta \equiv \frac{2\pi}{U}$. Chourdakis (2005) used this refined algorithm in pricing European options on equities based on the Carr and Madan (1999) pricing framework.

The FRFT was developed to efficiently compute the sum

$$f(v, \mathbf{h}, \zeta) = \sum_{u=1}^{U} e^{-2\pi\imath(u-1)(v-1)\zeta} h_u \qquad \text{for } v = 1, \ldots, U. \tag{6.8}$$

Thus, introducing the FRFT operator, we define the compact expression

$$\mathbf{f}(\mathbf{h}, \zeta) = \text{FRFT}\left[\mathbf{h}; \zeta\right].$$

Although, the parameter ζ is usually real-valued, it is not restricted to the set of $\mathbb{R}$. Obviously, the FRFT is strongly connected to the FFT and the IFFT. For example, by comparing equation (6.5) with (6.8), we have the equivalence

$$\text{IFFT}\left[\mathbf{h}\right] \equiv \frac{1}{U} \text{FRFT}\left[\mathbf{h}; -\frac{1}{U}\right].$$

The key insight to compute the FRFT in terms of the FFT and the IFFT algorithm, respectively, is to recognize that the product $2(u-1)(v-1)$ can be expressed as

[141] The fractional Fourier Transformation parameter ζ in this thesis corresponds to α in the original article of Bailey and Swarztrauber (1994).

$$(u-1)^2 + (v-1)^2 - (v-u)^2.$$

Inserting this relation into equation (6.8), subsequently doing some algebraic transformations and using the discrete version of the convolution theorem of Fourier Transformations[142], we are able to efficiently compute equation (6.8) with the help of both the FFT and the IFFT algorithm as follows[143]. Defining the vectors $\mathbf{p}$ and $\mathbf{q}$ with elements

$$p_u = \begin{cases} \frac{h_u}{a_u} & \text{for} \quad 1 \leq u \leq U \\ 0 & \text{for} \quad U < u \leq 2U, \end{cases}$$

and

$$q_u = \begin{cases} a_u & \text{for} \quad 1 \leq u \leq U \\ a_{(2U+2-u)} & \text{for} \quad U < u \leq 2U, \end{cases}$$

with

$$a_u = e^{\imath\pi\zeta(u-1)^2},$$

we compute first the raw transformation as

$$\hat{\mathbf{f}}(\mathbf{h},\zeta) = \text{IFFT}\left[\text{FFT}\left[\mathbf{p}\right] \odot \text{FFT}\left[\mathbf{q}\right]\right].$$

The last U elements in $\hat{\mathbf{f}}(\mathbf{h},\zeta)$ can be discarded due to the zero padding made in the vector $\mathbf{p}$. Thus, we store the first half of the vector $\hat{\mathbf{f}}(\mathbf{h},\zeta)$ in a new vector $\hat{\mathbf{f}}^{-}(\mathbf{h},\zeta)$. The FRFT is then

$$\mathbf{f}(\mathbf{h},\zeta) = \hat{\mathbf{f}}^{-}(\mathbf{h},\zeta) \odot \mathbf{a_{-u}}. \tag{6.9}$$

Obviously, by comparing the term inside the sum operator in equation (6.4) with the corresponding term inside the sum in equation (6.8) we have to establish the relation

$$\zeta = -\frac{\Delta\eta}{2\pi},$$

where both Δ and η can be chosen arbitrarily[144]. Thus, our general option-pricing formula (6.7), can be rewritten in terms of the FRFT as

[142] See Proposition 2.4.3.

[143] The detailed derivation of the FRFT algorithm is given in Bailey and Swarztrauber (1994).

[144] Note that the factor $\frac{1}{U}$ used in equation (6.4) is already included in Δ.

$$\begin{aligned}&\mathbf{V}(\mathbf{x}_t, t, T)\\ &= \frac{\Delta\, e^{(d-z_i)\boldsymbol{\alpha}}}{\pi} \odot \\ &\qquad \mathrm{Re}\left[e^{-\pi\imath(\mathbf{v}-1)\zeta} \odot \mathrm{FRFT}\left[e^{\imath\,\Delta\alpha_1(\mathbf{u}-\frac{1}{2})} \odot \hat{\mathbf{g}} \odot \boldsymbol{\psi}; -\frac{\Delta\eta}{2\pi}\right]\right].\end{aligned} \tag{6.10}$$

Although we have to compute two FFTs and one IFFT in order to obtain one FRFT, there is a substantial improvement due to the now independent choice of strike interval and integration domain, which saves in the end computer time. This fact becomes more important for the computation of characteristic functions for which no closed-form solutions exist and therefore the system of ODEs (2.40) and (2.41) must be solved numerically for each sampling point z_u.

6.3.3 Determination of the Optimal Parameters for the Numerical Scheme

As discussed in Lee (2004) and Lord and Kahl (2007), the choice of z_i, determining the specific contour in the complex plane used for the numerical integration routine is crucial in computing option prices. Lee (2004) finds that for different option payoff functions, for different strike values and driving processes, respectively, the optimal value of z_i, thus minimizing the numerical error, varies substantially[145]. Furthermore, the parameter ω concerning the truncation error is also of the utmost importance in a numerical option-pricing scheme. Thus, both parameters influence the accuracy of numerical solutions. This is illustrated in Figure 6.1 for zero-bond call options and the jump-enhanced models of Vasicek (1977) and Cox, Ingersoll and Ross (1985b)[146]. Obviously, setting ω too small results in a highly oscillating solution vector. On the other hand choosing ω too high, the absolute error of the numerical so-

[145] See Lee (2004) Table 2 and 3. The same observation is made in Lord and Kahl (2007), Figure 1.

[146] Both interest-rate models are enhanced with an exponentially distributed jump component. The coefficients for the characteristic function of the jump-enhanced Vasicek model are given in equations (8.6), (8.7), and (8.8). The particular coefficients in case of the jump-enhanced CIR model are given in equations (8.11), (8.12), and (8.13). A discussion of these models is given in Chapter 8.

lutions increase exponentially. The opposite statement holds for z_i. Therefore, these parameters should be chosen to avoid minimize both effects.

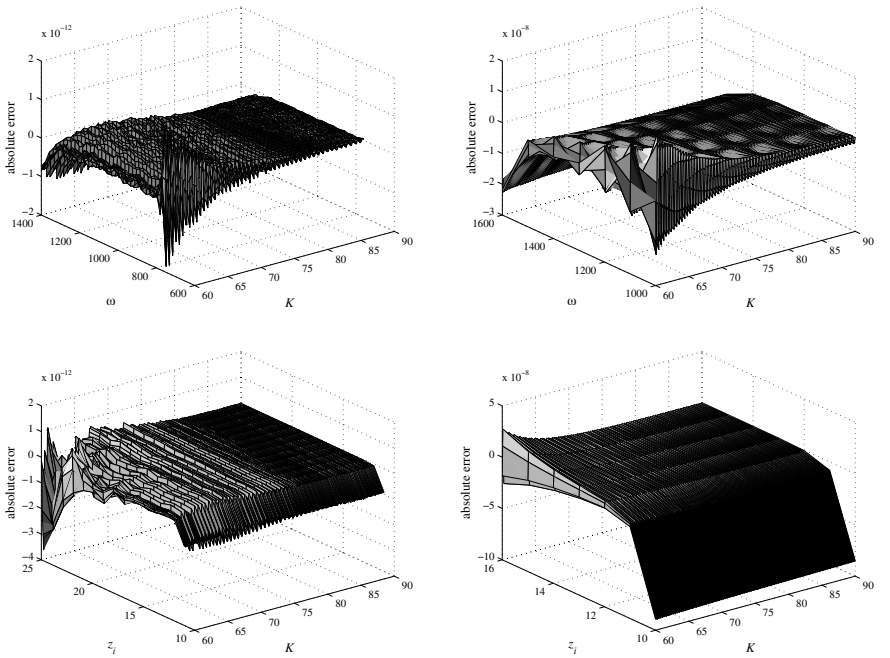

Fig. 6.1. Graphs in the first row depict absolute errors of 512 zero-bond call prices for alternating values of ω. In the second row, the particular errors are depicted for varying values of z_i. An exponential-jump version of the Vasicek (CIR) model is used in the left (right) column. The parameters are: $r_t = 0.05(0.03), \kappa = 0.4(0.3), \theta = 0.05(0.03), \sigma = 0.01(0.1), \eta = 0.005(0.005), \lambda = 2(2), \tau = 0.5(0.5), \hat{\tau} = 2(2)$.

Since we want to price a vector of option prices with the computation of one FRFT operation, thus considering one specific parameter setting for the entire strike range, we are interested in finding the optimal parameter setting for the pricing algorithm, (ω^*, z_i^*), which minimizes the overall numerical error in equation (6.10). Hence, we need a criterion which measures the cumulative error of both positive and negative deviations from the theoretical solutions. Consequently, we apply in the following analysis the root mean-squared error (RMSE), which is

$$\text{RMSE} = \sqrt{\frac{(\mathbf{V}^{Num} - \mathbf{V}^{True})'(\mathbf{V}^{Num} - \mathbf{V}^{True})}{U}}, \tag{6.11}$$

where $\mathbf{V}^{Num}$ denotes some numerical solution vector and $\mathbf{V}^{True}$ represents the corresponding vector of closed-form solutions. To give an idea of the error behavior of the FRFT pricing algorithm, we first compare quasi closed-form solutions computed with the QUADL integration routine in MATLAB[147] according to equation (6.1) and the corresponding values due to the FRFT algorithm as defined in equation (6.10) for a fixed number of 512 different strike rates. The particular natural logarithms of the RMSE for zero-bond call option prices are depicted in Figure 6.2. We make two remarkable observations. Firstly, for differing values of ω and z_i both models have a global minimum of the RMSE of computed option prices. Secondly, the logarithmic presentation of the RMSE implies a rapid and monotonic descent towards this minimum, starting with small values of ω and z_i[148]. In case of the jump-enhanced CIR model, the specific error-minimizing parameter couple is clearly evident according to the contour plot of the logarithmic RMSE given in the lower right graph of Figure 6.2. On the other hand, the particular contour plot of the logarithmic RMSE for zero-bond call options under the jump-enhanced Vasicek model also clearly indicates a region of parameter couples exhibiting approximately the same RMSE magnitude.

Consequently, we exploit this monotonic decrease of the RMSE to develop an algorithm, which is capable of finding an optimal parameter setting (ω^*, z_i^*) and simultaneously giving an estimate of the magnitude of errors of numerical solutions even when the closed-form solutions are not known. The technique we use for the approximation of the numerical error is based on the exponential decreasing of the mean-squared error between two successive parameter values in the numerical scheme. Thus, we define the approximate RMSE as

$$\text{RMSE}^a = \sqrt{\frac{(\mathbf{V}^{Num} - \mathbf{V}^{Num^{(+)}})'(\mathbf{V}^{Num} - \mathbf{V}^{Num^{(+)}})}{U}}, \tag{6.12}$$

where $\mathbf{V}^{Num}$ and $\mathbf{V}^{Num^{(+)}}$ denote numerical solutions of two successive parameter values, whether in ω or in z_i direction.

[147] This integration routine uses an adaptive Lobatto quadrature scheme. In the calculation of quasi closed-form solutions, we set its error tolerance to 10^{-15}.

[148] This phenomenon shows up for all interest-rate model/payoff combinations mentioned in this thesis.

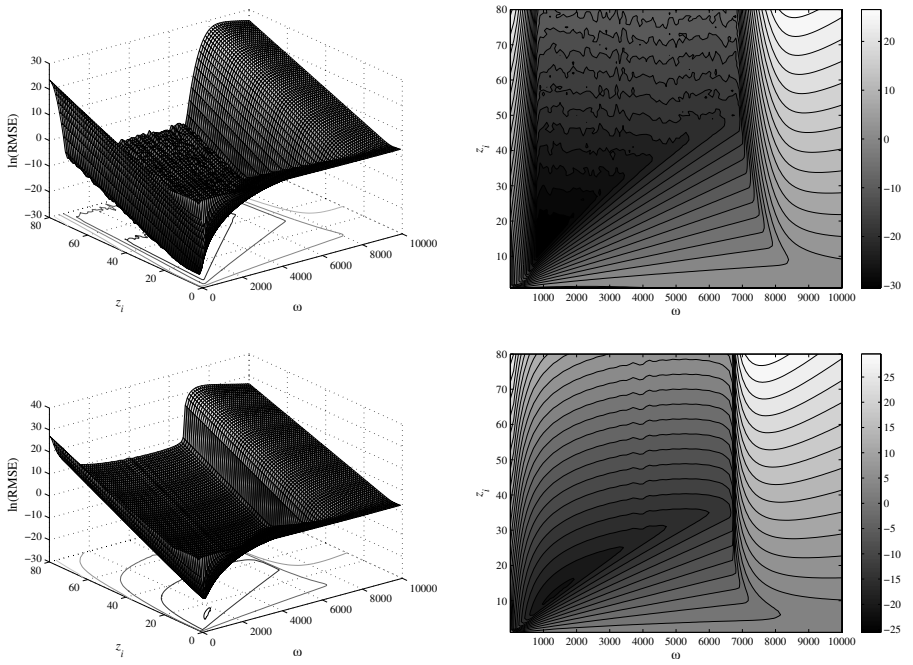

Fig. 6.2. Logarithmic RMSEs of 512 zero-bond call option prices. In the upper (lower) row the underlying interest rate is modeled by a jump-enhanced Vasicek (CIR) model. The parameters are: $r_t = 0.05(0.03), \kappa = 0.4(0.3), \theta = 0.05(0.03), \sigma = 0.01(0.1), \eta = 0.005(0.005), \lambda = 2(2), \tau = 0.5(0.5), \hat{\tau} = 2(2)$ and a strike range of $K \in [60, 90]$.

In Figure 6.3, differences of the logarithmic RMSE^a, for two successive parameter values of z_i, and the logarithmic RMSE according to equation (6.11) are depicted for zero-bond call prices for varying z_i values. Obviously, the approximate and exact RMSEs show nearly the same magnitude until the minimum RMSE is reached. Afterwards, the difference, still very small, becomes oscillating in case of the Vasicek model and experiences a decrease of its level in case of the CIR model, respectively. This characteristic behavior of the RMSE$^{\mathrm{a}}$ is used in our algorithm to find the optimal parameter couple (ω^*, z_i^*) and enables the formulation of an approximate error bound for the numerical solution vector.

As mentioned above, our algorithm to find the optimal parameter couple (ω^*, z_i^*) utilizes a steepest descent technique on the logarithm of the RMSE$^{\mathrm{a}}$.

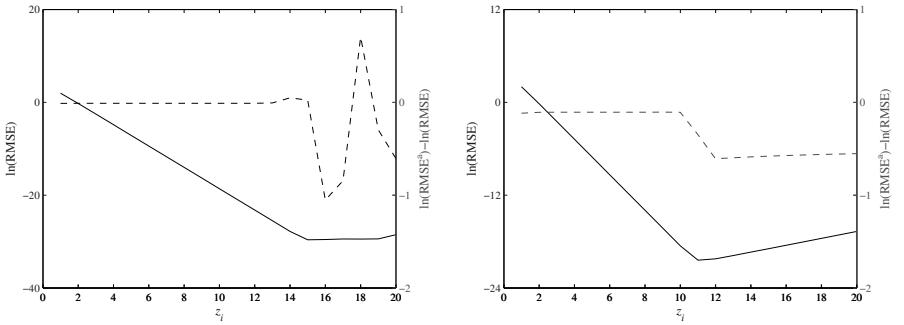

Fig. 6.3. The dashed line represents the difference of the logarithmic RMSE^a and the exact RMSE of 512 zero-bond call option prices and increasing values of z_i. Both graphs are drawn for $\omega = 1400$. The straight line depicts the logarithmic RMSE in dependence of z_i. The underlying model in the left (right) graph is a jump-enhanced Vasicek (CIR) model with parameters: $r_t = 0.05(0.03), \kappa = 0.4(0.3), \theta = 0.05(0.03), \sigma = 0.01(0.1), \eta = 0.005(0.005), \lambda = 2(2), \tau = 0.5(0.5), \hat{\tau} = 2(2)$ and a strike range of $K \in [60, 90]$.

Thus, initializing the algorithm, we first evaluate the numerical solution $\mathbf{V}^{Num}$ for some parameter values (ω^o, z_i^0)[149]. Subsequently, we compute two additional solution vectors for ascending parameter values in the direction of both ω and z_i which are then used to derive the particular first order finite differences. Afterwards, if the slope in ω direction is smaller than the one in z_i direction, thus more negative, the next numerical solution is computed with an exalted ω and vice versa. The next step in the numerical scheme is then again to obtain the necessary numerical solution vectors in order to derive the particular finite differences and so on. The algorithm aborts if the smallest value of $\ln(\text{RMSE}^a)$ is reached over some interval where the curve experienced its reversal point. In Figure 6.4, the paths with an initial value of $z_i^0 = 2$ and $\omega^0 = 10$ for the jump-enhanced Vasicek and CIR model are shown. Obviously, the algorithm finds for both interest-rate models the optimal parameter setting, which can be justified by the graphs in the left column of Figure 6.4. In case of the optimal parameter couple using the jump-enhanced Vasicek (CIR) model, we get a difference of exact and approximate RMSEs of 9.02924×10^{-14}

[149] Since we observe the steepest descent starting at the origin the initial value for z_i^0 and ω^0 has to be near the origin subject to the particular regularity conditions of the Fourier-transformed payoff function.

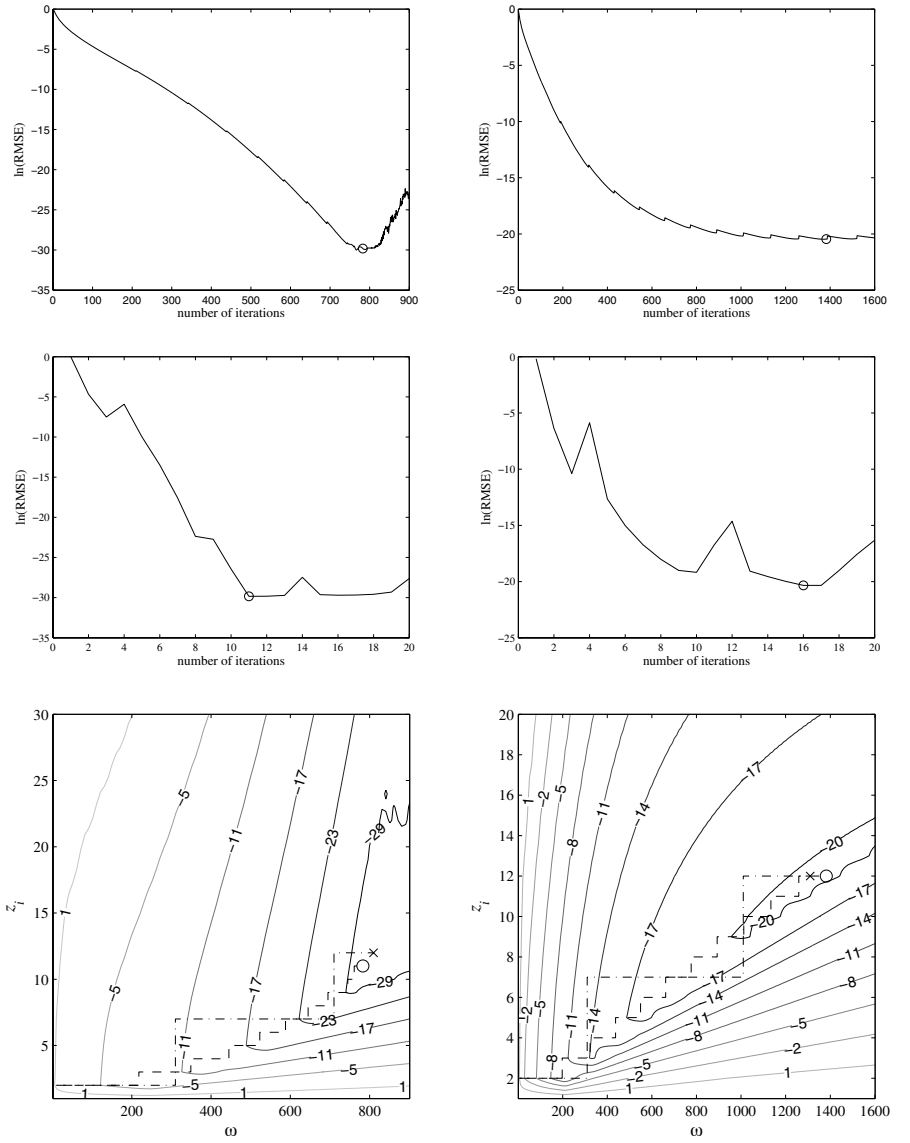

Fig. 6.4. Search for the optimal parameter couple (ω^*, z_i^*). In the first (second) column particular graphs are shown for the Vasicek (CIR) model with the data used in Figure 6.3. In the first row, the particular ln(RMSE) is depicted for the search algorithm with increments $(\Delta\omega, \Delta z_i) = (1, 1)$. In the second row the same search is made with increments (100, 5). In the third row the dashed (dash-dotted) line denotes the particular search path for small (high) increment values, where the optimal choice is marked by a circle and cross, respectively.

(8.27740×10^{-10}), whereas the exact RMSE is 1.11766×10^{-13} (1.30453×10^{-9}). Thus, we have in both models a difference which is of smaller order than the effective error according to the RMSE. Consequently, the RMSE^a gives a good prediction for the corresponding exact value, which justifies the application of the approximate RMSE. In the first row of Figure 6.4, we used very small increments for the search of the optimal parameter couple to give a detailed impression of the search path and the particular logarithmic RMSE. According to the graphs in the second row of Figure 6.4, a comparable result is achieved by running the algorithm with higher increments[150]. However, due to the reduced number of iterations, the search algorithm with high increments is in case of the jump-enhanced Vasicek (CIR) model up to 71 (86) times faster. Dealing with a characteristic function known in closed form together with a FRFT-based pricing algorithm, the search takes only a second at all even for small increments. Thus, if the general characteristic function is known in closed form, the step-size does not matter. However, if values of the general characteristic need to be determined numerically via a Runge-Kutta algorithm, we usually set the increments high enough to keep the overall number of iterations small.

Finally, we use the RMSE^a to derive an upper error bound for the numerical solutions contained in $\mathbf{V}^{Num}$. The first step in deriving this particular error bound is to consider a hypothetical solution vector $\mathbf{V}^{Num}$, where all elements equal their true solutions except the result given in the first position of the solution vector, namely V_1^{Num}. Without loss of generality, we assume the numerical error of this particular option price to be of magnitude $|a|$. Therefore, solving equation (6.11) in this special case gives

$$a = \text{RMSE}\sqrt{U}. \tag{6.13}$$

Additionally, we are also able to state the inequality

$$\sqrt{(\mathbf{V}^{Num} - \mathbf{V}^{True})'(\mathbf{V}^{Num} - \mathbf{V}^{True})} \geq |V_v^{Num} - V_v^{True}|, \tag{6.14}$$

to hold for every element of the numerical solution vector $\mathbf{V}^{Num}$. According to equation (6.13), the RMSE scaled by some constant $\sqrt{U}$ states the value

[150] The second run of the algorithm, with higher increments, gives an absolute error for the optimal parameter couple (ω^*, z_i^*) for zero-bond calls under the jump-enhanced Vasicek (CIR) model of 1.13911×10^{-13} (1.46601×10^{-9}).

of the maximum attainable error. Furthermore, this result together with the inequality in (6.14) generally implies that the absolute error of one particular numerical solution V_v^{Num} cannot exceed the absolute value $|a|$. Therefore, the RMSE can be used in formulating a boundary for the highest possible error. Consequently, we use the quantity RMSE^{a}, scaled by some constant $\sqrt{U}$, as a conservative upper error bound for the results generated by the pricing algorithm.