



Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational  
Mathematics (IMACM)

Preprint BUW-IMACM 25/02

Anna Clevenhaus, Claudia Totzeck and Matthias Ehrhardt

**A Space Mapping approach for the calibration of financial  
models with the application to the Heston model**

January 21, 2025

<http://www.imacm.uni-wuppertal.de>

# A Space Mapping approach for the calibration of financial models with the application to the Heston model

Anna Clevenhaus<sup>1</sup>, Claudia Totzeck<sup>2</sup> and Matthias Ehrhardt<sup>1</sup>

<sup>1</sup> Applied and Computational Mathematics, University of Wuppertal, Germany

<sup>2</sup> Continuous Optimization, University of Wuppertal, Germany

## Abstract

We present a novel approach for parameter calibration of the Heston model for pricing an Asian put option, namely space mapping. Since few parameters of the Heston model can be directly extracted from real market data, calibration to real market data is implicit and therefore a challenging task. In addition, some of the parameters in the model are non-linear, which makes it difficult to find the global minimum of the optimization problem within the calibration. Our approach is based on the idea of space mapping, exploiting the residuum of a coarse surrogate model that allows optimization and a fine model that needs to be calibrated. In our case, the pricing of an Asian option using the Heston model SDE is the fine model, and the surrogate is chosen to be the Heston model PDE pricing a European option. We formally derive a gradient descent algorithm for the PDE constrained calibration model using well-known techniques from optimization with PDEs. Our main goal is to provide evidence that the space mapping approach can be useful in financial calibration tasks. Numerical results underline the feasibility of our approach.

## 1 Introduction

Financial models aim to approximate the real market behavior of various underlyings. Since most model parameters are implicit in the real market data, calibrating the model parameters to fit the real market data is challenging [7, 13, 19, 21, 22, 25]. In this article, we consider the well-known Heston model [11], a two-dimensional stochastic differential equation (SDE) that allows to simulate the behavior of stock prices. In particular, for a given variance of the option we calibrate four implicit parameters – the volatility of the variance, the mean inversion rate, the long-term mean and the correlation factor of the Brownian motions in the model equations for the asset and the variance – to the market data. Our goal is to provide a proof-of-concept, so we choose the simplest model with constant coefficients. However, we are aware that the Heston model can be extended in

various ways, e.g. by considering time-dependent parameters [22] or by including additional processes, which leads to an increase in the difficulty of the calibration process [25].

Various calibration techniques for the Heston model are available in the literature. Indeed, in the setting of constant parameters and for very specific use cases, there are approaches based on the closed-form valuation formula of the Heston model [22, 7]. These are fast and provide information about the global minimum. For more general cases, the stochastic nature of the Heston model allows for Monte Carlo optimization methods [25], which can also be used to calibrate the stock price and variance. The Monte Carlo theory is well established, but the approaches are computationally expensive and do not provide information about global or local minima. Recently, calibration approaches using neural networks, deep learning strategies as well as parallel GPU implementation of the Heston model have been proposed [13, 19, 20, 18, 9]. The networks must be trained individually for each model, and training requires appropriate data. Again, there is no information about global or local minima. An advantage is that once the neural networks are trained, they can be evaluated quickly.

Our contribution to this line of research is a calibration algorithm for the Heston model that is independent of a specific characteristic function and easily extendable to time-dependent parameters. The core of the algorithm is based on the space mapping [2], an iterative procedure that minimizes the residuum of a fine and a coarse model. In fact, to calibrate the parameters of the fine model, the coarse model is optimized and the fine model is only evaluated. To demonstrate the proposed approach, we calibrate the Heston SDE model for pricing Asian put options (fine model) to real market data. The calibration uses only one given market price. In our setting, the coarse model is chosen because it is the computationally less expensive Heston model for pricing the European put option, which is known to be a good approximation for the Asian option in the case of short maturities. This is because the European option is a limiting case for the Asian option. In more detail, while we aim to calibrate the parameters of the SDE for the Asian put option, we will optimize the deterministic PDE model of the European option, which can be solved using techniques from optimization with PDE [12, 27]. In each iteration of the calibration process, we will evaluate the SDE to compute the residuum of the two models. To our knowledge, there is only one paper dealing with the calibration of Asian options under the Heston model, where Khalife et al. [15] calibrate the Heston model using artificial intelligence and the approximation is done via the European option as well.

The proposed calibration is based on *aggressive space mapping* (ASM) [8]. To iteratively update the parameters of the fine model, a PDE-constrained optimization problem is considered at the level of the coarse model. We measure the difference between the fair price given by the numerical solution of our model and the reference data, the subsequent market data, in the cost functional. The PDE-constrained optimization problem is solved using a gradient descent method. For this purpose, we formally derive an adjoint-based gradient descent algorithm for the Heston PDE model. Note that gradient descent algorithms have previously been used implicitly in the context of neural network approximations for the Heston model [20].

As mentioned above, our main goal is to provide a proof-of-concept that the space

mapping approach can be useful for calibration tasks in financial research. This claim is supported by the numerical results. However, we are aware that there is much room for improvement, for example an extension to time-dependent parameters, see [5] for details on the PDE-constrained optimization problem.

The article has the following structure. In Section 2, we present the mathematical formulation of the Heston SDE model and its log-transformed form for pricing an Asian put option. We also discuss the PDE formulation of the Heston model for pricing a European put option. In Section 3, we then introduce the corresponding optimization problems and present details of the space mapping approach. This includes the formal derivation of the adjoint of the log-transformed Heston model as well as the identification of the gradient which is used in the descent algorithm. We then discretize the SDE model and the PDE calibration method in Section 4. Numerical results to illustrate the feasibility of space mapping in finance are presented in Section 5. Finally, in Section 6 the paper ends with a conclusion and an outlook.

## 2 The Heston SDE model and its PDE approximation

The Heston model was proposed by Heston in 1993 [11] and describes the dynamics of the underlying asset and its variance by a two-dimensional SDE. It is an extension of the well-known Black-Scholes model [3]. Let  $S \in \mathbb{R}^+$  denote the asset,  $r \in \mathbb{R}^{>0}$  the risk-free interest rate,  $\sigma_S \in \mathbb{R}^{>0}$  the volatility and  $W_t^S$  the Brownian motion. Then the Black-Scholes model describes the dynamics of  $S$  as a geometric Brownian motion process given by

$$dS_t = rS_t dt + \sigma_S S_t dW_t^S, \quad t \in [0, T], \quad S_0 > 0, \quad (1)$$

where  $T$  is the maturity.

Heston introduced a second stochastic process to model more complex volatility behavior. Let  $\nu \in \mathbb{R}^+$  denote the variance, set  $\sigma_S = \sqrt{\nu}$ , and assuming that the volatility follows an Ornstein-Uhlenbeck process, he includes a Cox-Ingersoll (Ross) process for the variance. This leads to Heston's SDE model under the risk-neutral measure given by

$$\begin{cases} dS_t = (r - q)S_t dt + \sqrt{\nu_t}S_t dW_t^S, & S_0 > 0, \\ d\nu_t = \kappa_\nu(\mu_\nu - \nu_t) dt + \sigma_\nu\sqrt{\nu_t} dW_t^\nu, & \nu_0 > 0, \end{cases} \quad (2)$$

where  $q \in \mathbb{R}^{>0}$  is the continuous dividend rate,  $\kappa_\nu \in \mathbb{R}^{>0}$  is the mean reversion rate,  $\mu_\nu \in \mathbb{R}^{>0}$  is the long term mean, and  $\sigma_\nu \in \mathbb{R}^{>0}$  is the volatility of the variance. The Brownian motions  $W_t^S$  and  $W_t^\nu$  are correlated by the constant  $\rho \in [-1, 1]$ , cf. [11]. To ensure that the square root of (2) is positive, the Feller condition  $2\kappa_\nu\mu_\nu \geq \sigma_\nu^2$  must be satisfied. Otherwise, computational problems arise due to a complex variance. The Heston SDE representation is used to price Asian put options with the payoff function

$$\phi_S(S, T) = \max(K - A_S(0, T)), \quad (3)$$

where  $K$  is the predetermined strike price and

$$A_S(0, T) = \frac{1}{T} \int_0^T S_t dt. \quad (4)$$

## 2.1 The fine model

To zoom into the particularly interesting price range near  $K$ , we use the variable transformation  $x = \log(S)$  for the asset. We get the log-transformed Heston SDE

$$\begin{cases} dx_t = (r - q - \frac{1}{2}\nu_t) dt + \sqrt{\nu_t} dW_t^S, & S_0 > 0, \\ d\nu_t = \kappa_\nu(\mu_\nu - \nu_t) dt + \sigma_\nu \sqrt{\nu_t} dW_t^\nu, & \nu_0 > 0, \\ dW_t^x dW_t^\nu = \rho dt \end{cases} \quad (5)$$

on the semi-unbounded domain  $x \in \mathbb{R}$ ,  $\nu \geq 0$ ,  $0 \leq t \leq T$  with the transformed payoff function

$$\phi(x, T) = \max\left(K - A(0, T)\right) = \max\left(K - \frac{1}{T} \int_0^T \exp(x_t) dt\right). \quad (6)$$

The payoff condition holds at the maturity and we are interested in the fair price  $V_f(x, t)$  today at  $t = 0$  given by

$$V_f(x, 0) = \exp(-rT) \mathbb{E}\left(\phi(x, T)\right) \quad (7)$$

using the discounted expectation value. This model will be used as the fine model in the space mapping approach discussed in Section 3.

## 2.2 The coarse model

As we will discuss in more detail below, the theory of space mapping requires that the coarse model be a reasonable approximation of the fine model that can be easily optimized. Since the European plain vanilla option is a limiting case for the Asian put option, we choose it as the coarse model. Furthermore, we use the PDE formulation of the Heston model for the European plain vanilla option because we can optimize it using methods from optimization with PDE constraints.

Note that the payoff condition (6) holds at maturity and thus yields a terminal condition. To obtain an initial value problem, we reverse the time  $\tau = T - t$ . Then, the Heston PDE under a risk-neutral measure is derived using Kolmogorov's backward equation. For the fair price of an option  $V(x, \nu, \tau)$  it reads

$$V_\tau = \frac{\nu}{2} V_{xx} + \frac{1}{2} \sigma_\nu^2 \nu V_{\nu\nu} + (r - q - \frac{\nu}{2}) V_x + \kappa_\nu (\mu_\nu - \nu) V_\nu + \sigma_\nu \nu \rho V_{x\nu} - rV, \quad (8)$$

and is supplemented with the initial condition

$$V(x, \nu, \tau) = \max\left(K - \exp(x), 0\right), \quad (9)$$

and appropriate boundary conditions. We choose to use the boundary conditions proposed by Heston and apply the log-transformation to obtain

$$V(x, \nu, \tau) \sim K \exp(-r\tau), \quad \text{for } x \rightarrow -\infty, \quad (10)$$

$$V(x, \nu, \tau) = 0, \quad \text{for } x \rightarrow \infty, \quad (11)$$

$$V(x, \nu, \tau) \sim K \exp(-r\tau), \quad \text{for } \nu \rightarrow \infty. \quad (12)$$

Note that the parabolic PDE (8) degenerates to a first-order hyperbolic PDE at  $\nu = 0$ , therefore we need to consider the Fichera theory [4, 17] to assess whether it is necessary to provide an analytic boundary condition at  $\nu = 0$  or not. From the Fichera condition at  $\nu = 0$  given by

$$b(\nu) = \kappa_\nu(\mu_\nu - \nu) - \frac{1}{2}\sigma_\nu^2 \quad (13)$$

we get a case dependency. At  $\nu = 0$  we get an outflow boundary if  $\lim_{\nu \rightarrow 0^+} b(\nu) \geq 0$ , otherwise we get an inflow boundary. The condition for an outflow boundary (13) is equivalent to the Feller condition  $2\kappa_\nu\mu_\nu \geq \sigma_\nu^2$ , which motivates us to assume in the following that the Feller condition holds, so that we do not have to supply an analytical boundary condition at  $\nu = 0$ . At this point we want to mention that we need a numerical boundary condition to complete the scheme, which will be discussed later in Section 3.1.

### 3 The Space Mapping Approach

Below we summarize the main ideas of the space mapping approach. For more details we recommend [2, 8, 26, 23]. The purpose of space mapping is to optimize (or calibrate) an accurate and computationally expensive model (fine) that cannot be optimized using a surrogate model (coarse) for which efficient optimization algorithms are available.

In our case, the accurate model is the Heston SDE for Asian option pricing, and the coarse model is the Heston PDE for European option pricing, as discussed above. Both models return a single price for a predefined asset  $\tilde{S}$  and a variance  $\tilde{\nu}$ . The predefined asset is part of the contract, but  $\tilde{\nu}$  must be determined implicitly from the market data or, as in our case, predefined by a guess. Together with the other unknown model parameters, we want to calibrate  $\xi = (\sigma_\nu, \rho, \kappa_\nu, \mu_\nu)$ . Note that we can take advantage of the fact that the parameters we want to calibrate are present in both models.

Although we expect the optimal values of the parameters to be different for the two models, the space mapping technique helps us to calibrate the parameters of the fine model while only optimizing the coarse model. In the following, we distinguish between  $\xi_f$ , the parameter vector for the fine model, and  $\xi_c$ , the parameter vector for the coarse model. Similarly, we denote the option price of the fine model by  $V_f$  and the option price of the coarse model by  $V_c$ .

Since we are using real market data  $V_{\text{data}}$  as ground truth for the calibration, we define the cost functional as follows

$$J(V; V_{\text{data}}) = \frac{1}{2} \int_0^T \|V - V_{\text{data}}\|^2 d\tau. \quad (14)$$

In the following, we want to approximate the solution of the fine calibration problem

$$\min_{\xi_f \in \tilde{X}_f} J(V_f; V_{\text{data}}) \quad \text{subject to} \quad V_f(x, 0) = \exp(-rT)\mathbb{E}(\phi(x, T)), \quad (15)$$

where  $x$  is the solution of the log-transformed Heston SDE (5). Note that we do not solve the fine optimization problem, but only evaluate the cost functional for given parameter sets  $\xi_f$  during the space mapping algorithm.

The optimization problem, which aims to calibrate the coarse model for given real market data, is given by

$$\min_{\xi_c \in X_c} J(V_c; V_{\text{data}}) \quad \text{subject to } V_c \text{ solves (8)}. \quad (16)$$

In the following we assume that both problems (15) and (16) admit a unique solution with a unique minimizer.

The solution of the coarse calibration problem will be our initial guess for the optimal parameter set of the fine model. However, in the space mapping approach we want to iteratively improve the parameter values of the fine model and we exploit the approximation properties of the coarse model. Let's assume that there exists a so-called *space mapping function*  $s: X_f \rightarrow X_c$ ,  $\xi_f \mapsto s(\xi_f)$ , which satisfies

$$s(\xi_f) := \operatorname{argmin}_{\xi_c \in X_c} r(V_c(\xi_c), V_f(\xi_f)), \quad (17)$$

for some misalignment function  $r: Y \times Y \rightarrow \mathbb{R}$ .

Assuming that the coarse model is a good approximation of the fine model, and choosing  $J$  as the misalignment function, we expect the following condition to be satisfied

$$\begin{aligned} s(\xi_f^*) &= \operatorname{argmin}_{\xi_c \in X_c} J(V_c(\xi_c); V_f(\xi_f^*)) \\ &\approx \operatorname{argmin}_{\xi_c \in X_c} J(V_c(\xi_c); V_c(\xi_c^*)) \\ &\approx \operatorname{argmin}_{\xi_c \in X_c} J(V_c(\xi_c); V_{\text{data}}) = \xi_c^*. \end{aligned}$$

The underlying assumption is, that the optimal states  $V_f$  and  $V_c$  are both good approximations of the ground truth  $V_{\text{data}}$ , each for the respective model.

The strategy of the space mapping algorithm is to solve

$$s(\xi_f^*) - \xi_c^* = 0.$$

Note that we will not approximate the entire function  $s$ , but only evaluate it along the iterations. Approximating  $s$  entirely is a much harder and probably ill-posed task.

For the numerical results we use a simplified version of the Aggressive Space Mapping (ASM) algorithm [23], in fact by the linearity of the state problems (5) and (8) we can approximate the Jacobian of the space mapping function by the identity [1] and thus obtain the Algorithm 1.

**Algorithm 1:** The simplified Aggressive Space Mapping (ASM) algorithm

---

**Result:** optimized  $\xi_f$   
 $\xi_f^0 = \xi_c^* = \operatorname{argmin}_{\xi_c \in X_c} \|c(\xi_c) - V\|;$   
**for**  $k = 0, 1, \dots$  **do**  
    1) evaluate space mapping function  $s(\xi_f^k)$  by ;  
        (a) evaluate fine model (7) to obtain  $V_f$  using  $\xi_f^k$   
        (b) Perform a coarse model optimization  
             $s(\xi_f^k) = \operatorname{argmin}_{\xi_c \in X_c} J(V_c(\xi_c); V_f(\xi_f^k))$   
    2) Compute  $h^k = -(s(\xi_f^k) - \xi_c^*)$  for update  $h_k$   
    4) Update control  $\xi_f^{k+1} = \xi_f^k + h^k$  using the projected Armijo rule to restrict  $\xi_f^{k+1}$  to the boundary  
    while  $\|s(\xi_f^k) - \xi_c^*\| > \text{tolerance};$   
**end**

---

### 3.1 Optimal Control of the Coarse Model

In the following, we formally derive the first-order optimality system of the coarse calibration problem that we solve in (b). For notational convenience, we use  $J(V_c(\xi_c); V_{\text{data}})$  in the derivation. Our formal derivation is based on a Lagrangian. Specifically, we formally derive a gradient-based algorithm using a Lagrangian approach to solve

$$\xi_c^* := \operatorname{argmin}_{\xi_c \in X_c} J(V_c, V_{\text{data}}) \text{ subject to (8)} \quad (18)$$

for a given data set  $V_{\text{data}}$ . We denote the Lagrange multipliers by  $\psi = (\varphi, \varphi^a, \varphi^b, \varphi^c, \varphi^d)$ , set  $\Omega = (0, \nu_{\max}) \times (x_{\min}, x_{\max})$  and partition the boundary  $\partial\Omega$  into

$$\begin{aligned} \Gamma_a &= \partial\Omega \cap \{x = -\infty\}, & \Gamma_b &= \partial\Omega \cap \{x = \infty\}, \\ \Gamma_c &= \partial\Omega \cap \{\nu = 0\}, & \Gamma_d &= \partial\Omega \cap \{\nu = \infty\}. \end{aligned}$$

From the log-transformed Heston PDE (8), we define

$$A = \frac{1}{2}\nu \begin{pmatrix} \sigma_\nu^2 & \sigma_\nu\rho \\ \sigma_\nu\rho & 1 \end{pmatrix}, \quad b = \begin{pmatrix} \kappa_\nu(\mu_\nu - \nu) - \frac{1}{2}\sigma_\nu^2 \\ r - q - \frac{\nu}{2} - \frac{1}{2}\sigma_\nu\rho \end{pmatrix}$$

and rewrite it to its divergent form

$$V_\tau - \nabla \cdot A \nabla V - b \cdot \nabla V + rV = 0.$$

The *constraint operator*  $e$  in the Lagrangian is implicitly defined by

$$\begin{aligned} \langle e(V, \xi), \psi \rangle &= \int_0^T \int_{\tilde{\Omega}} V \left[ -\varphi_\tau - \nabla \cdot A^\top \nabla \varphi + b \cdot \nabla \varphi + (r + \nabla \cdot b) \varphi \right] dz \\ &\quad + \int_{\partial\Omega} [(A^\top \nabla \varphi) \cdot \vec{n} - (b \cdot \vec{n}) \varphi] V ds - \int_{\partial\Omega} (A \nabla V) \cdot \vec{n} \varphi ds d\tau \\ &\quad + \int_0^T \int_{\Gamma_a} [V - \exp(-r\tau)] \varphi^a ds d\tau + \int_0^T \int_{\Gamma_b} V \varphi^b ds d\tau \\ &\quad + \int_0^T \int_{\Gamma_d} [V - \exp(-r\tau)] \varphi^d ds d\tau, \end{aligned} \quad (19)$$



where  $\tilde{\Omega} = \Omega \cap \Gamma_c$ , since no boundary condition is required at  $\Gamma_c$ .

Following the approach of the authors [5], we obtain the adjoint equation

$$\begin{aligned} \varphi_\tau + \frac{1}{2}\nu\sigma_\nu^2\varphi_{\nu\nu} + \nu\sigma_\nu\rho\varphi_{x\nu} + \frac{1}{2}\nu\varphi_{xx} + (\sigma_\nu^2 - \kappa_\nu(\mu_\nu - \nu))\varphi_\nu \\ + (q - r + \frac{\nu}{2} + \sigma_\nu\rho)\varphi_x + (\kappa_\nu - r)\varphi = -(V - V_{\text{data}}) \quad \text{on } \Omega \end{aligned} \quad (20)$$

with terminal condition  $\varphi(T) = 0$  and  $\varphi = 0$  on the boundaries  $\Gamma_a$ ,  $\Gamma_b$  and  $\Gamma_d$  and the outflow boundary at  $\nu = 0$ . For a more detailed derivation of the adjoint formulation, see [5].

We compute the optimality condition by setting  $d_\xi L(V, \xi, \psi) = 0$ . Since the boundaries  $\Gamma_a$ ,  $\Gamma_b$  and  $\Gamma_d$  are zero, we focus on  $\tilde{\Omega}$ . In the following, the derivatives with respect to the different parameters are given separately

$$\begin{aligned} d_{\sigma_\nu} \langle e(V, \xi), \psi \rangle &= \int_0^T \int_{\tilde{\Omega}} V \left[ -\sigma_\nu \nu \varphi_{\nu\nu} - 2\sigma_\nu \varphi_\nu - \rho \varphi_x - \rho \nu \varphi_{x\nu} \right] dz d\tau, \\ d_\rho \langle e(V, \xi), \psi \rangle &= \int_0^T \int_{\tilde{\Omega}} V \left[ -\sigma_\nu \varphi_x - \sigma_\nu \nu \varphi_{x\nu} \right] dz d\tau, \\ d_{\kappa_\nu} \langle e(V, \xi), \psi \rangle &= \int_0^T \int_{\tilde{\Omega}} V \left[ (\mu_\nu - \nu) \varphi_\nu - \varphi \right] dz d\tau, \\ d_{\mu_\nu} \langle e(V, \xi), \psi \rangle &= \int_0^T \int_{\tilde{\Omega}} \kappa_\nu V \varphi_\nu dz d\tau. \end{aligned}$$

Note that  $d_\xi L(V, \xi, \psi)[h_\xi] = 0$  must hold for arbitrary directions  $h_\xi$ . Therefore, we can read off the gradient from the above expressions. We solve the state equation for the Heston model with a constant control variable  $\xi$  (8) for a given initial parameter set  $\xi_0$ . With the state solution at hand, we can compute the corresponding adjoint equation (20) backwards in time. The state and adjoint solutions allow us to compute the gradient for the descent step.

We ensure that  $\kappa_\nu$ ,  $\mu_\nu$ ,  $\sigma_\nu$ , and  $\rho$  are within their parameter bounds, and that the Feller condition is satisfied by using the *projected Armijo rule* [27]. In the projected Armijo rule, we choose the maximum  $\sigma_k \in \{1, 1/2, 1/4, \dots\}$  for which

$$f(\mathcal{P}(\xi_k - \sigma_k \nabla f(\xi_k))) - f(\xi_k) \leq -\frac{\gamma}{\sigma_k} \|\mathcal{P}(\xi_k - \sigma_k \nabla f(\xi_k)) - \xi_k\|_2^2.$$

Here,  $\gamma \in (0, 1)$  is a numeric constant, we will use the default choice with  $\gamma = 10^{-4}$ .

## 4 Numerical Scheme

Since this is a proof-of-concept, simple and well-known spatial and temporal discretization methods are used to illustrate our approach for discretizing both the SDE and the PDE.

### 4.1 The Fine Model: Heston SDE

For the fine model, we consider a Monte Carlo simulation with variance reduction. We consider the log-transformed SDE

$$\begin{cases} dx_t = (r - q - \frac{1}{2}\nu_t) dt + \sqrt{\nu_t} dW_t^S, & S_0 > 0, \\ d\nu_t = \kappa_\nu(\mu_\nu - \nu_t) dt + \sigma_\nu \sqrt{\nu_t} dW_t^\nu, & \nu_0 > 0, \\ dW_t^x dW_t^\nu = \rho dt \end{cases} \quad (21)$$

and use the well-known Euler-Maruyama scheme to discretize the SDE. Therefore we discretize the time uniformly  $t_k = k\Delta t$ , where  $\Delta t = T/N_t$  and  $k = 0, \dots, N_t$ . Given two independent normal distributed random variables for each time step  $\omega_{1,k}$  and  $\omega_{2,k}$ , we generate

$$\begin{aligned} \omega_{x,k} &= \omega_{1,k}, \\ \omega_{\nu,k} &= \rho\omega_{x,k} + \sqrt{1 - \rho^2}\omega_{2,k}, \end{aligned} \quad (22)$$

and obtain the discretized scheme

$$\begin{cases} x_{k+1} = x_k + (r - q - \frac{1}{2}\nu_t) \Delta t + \sqrt{\nu_t} \sqrt{\Delta t} \omega_{x,k} \\ \nu_{t+1} = \nu_t + \kappa_\nu(\mu_\nu - \nu_t) \Delta t + \sigma_\nu \sqrt{\nu_t} \sqrt{\Delta t} \omega_{\nu,k} \end{cases} \quad (23)$$

To reduce the variance, we use antithetic variables [10] to obtain the so-called antithetic path  $x^-$ . Since we use a uniform spacing for the time discretization, the discrete integral  $A(0, T)$  of the payoff function is approximated by

$$A(0, T) = \frac{1}{T} \int_0^T \exp(x_t) dt \approx \frac{1}{T} \frac{T}{N_t} \sum_{k=1}^{N_t} \exp(x_k) = \frac{1}{N_t} \sum_{k=1}^{N_t} \exp(x_k) = \text{mean}(x). \quad (24)$$

If  $N_p$  is the number of paths generated for the Heston model, we get the payoff for a path  $p$

$$\phi_p(x, x^-, T) = \frac{1}{2} \left( \max(K - \text{mean}(x), 0) + \max(K - \text{mean}(x^-), 0) \right) \quad (25)$$

and the fair price of the option is given by

$$\exp(-rT) \frac{1}{N_p} \sum_{i=1}^{N_p} \phi_{p_i}(x, x^-, T). \quad (26)$$

### 4.2 The Coarse Model: Heston PDE

Before discretizing the Heston PDE, we perform a domain truncation and introduce a closure boundary condition at  $\nu = 0$  for the Heston and its adjoint formulation. Since we assume that the Feller condition holds, according to Fichera theory we have a pure outflow boundary at  $\nu = 0$  and no need for an analytical boundary condition, neither for the Heston model nor for its adjoint formulation. However, we do need a closure condition at this boundary for the discretized PDE. As discussed in [6, 17], we follow Heston's

approach and use the reduced hyperbolic formulation of the Heston PDE and its adjoint. At  $\Gamma_c$ , we obtain for the log-transformed normalized PDE and similarly for the adjoint formulation

$$\varphi_\tau + (\sigma_\nu^2 - \kappa_\nu \mu_\nu) \varphi_\nu + (q - r + \sigma_\nu \rho) \varphi_x + (\kappa_\nu - r) \varphi = -(V - V_{\text{data}}). \quad (27)$$

We obtain a rectangular grid by truncating the domain and introducing grid points. We consider uniform meshes in each direction and get  $x_i = x_{\min} + i\Delta_x$  for  $i = 0, \dots, N_x$  with  $\Delta_x = (x_{\max} - x_{\min})/N_x$  and  $\nu_j = j\Delta_\nu$  for  $j = 0, \dots, N_\nu$  with  $\Delta_\nu = \nu_{\max}/N_\nu$  for the spatial direction, and  $\tau_k = k\Delta_\tau$  for  $k = 0, \dots, N_\tau$  with  $\Delta_\tau = T/N_\tau$  for the time direction.

For the time discretization we use the Hundsdorfer-Verwer scheme [14], a well-known *alternating direction implicit* (ADI) method that is able to handle mixed derivative terms. Within the Hundsdorfer-Verwer scheme,  $\theta$  is introduced as a classification measure similar to the  $\theta$  method and is of second order for any choice of  $\theta$ . Since the log-transformed Heston PDE and its adjoint formulation are second-order PDEs, they are discretized similarly. Therefore, we introduce a general second-order PDE formulation

$$u_\tau + a_{11}u_{\nu\nu} + 2a_{12}u_{x\nu} + a_{22}u_{xx} + b_1u_\nu + b_2u_x + cu = 0. \quad (28)$$

As a first step, we split the PDE operator into three operators

$$\mathcal{F}(\tau) = \mathcal{F}_0(\tau) + \mathcal{F}_1(\tau) + \mathcal{F}_2(\tau). \quad (29)$$

Each operator deals with a spatial direction,  $F_0$  includes the mixed derivative terms,  $F_1$  includes the derivatives in  $x$ -direction, and similarly  $F_2$  includes the derivatives with respect to  $\nu$ . Since the term  $cu$  is independent, it is split in half and added to both  $F_1$  and  $F_2$ . We get the splitting

$$\begin{aligned} \mathcal{F}_0(\tau) &= 2a_{12}D_{x\nu}, \\ \mathcal{F}_1(\tau) &= b_2D_x + a_{22}D_{xx} - \frac{1}{2}cI, \\ \mathcal{F}_2(\tau) &= b_1D_\nu + a_{11}D_{\nu\nu} - \frac{1}{2}cI, \end{aligned}$$

where  $D_x$  describes the discretization matrix of the first derivative w.r.t.  $x$  and correspondingly  $D_{xx}$  of the second derivative w.r.t.  $x$ ,  $D_\nu$  and  $D_{\nu\nu}$  of the first and second derivatives w.r.t.  $\nu$ ,  $I$  denotes the identity matrix. We use central finite difference stencils to derive the corresponding matrices.

Let  $u_k^{i,j} \approx u(x_i, \nu_j, \tau_k)$  and simplify  $u_k \approx u(x, \nu, \tau_k)$ . In each time step, we have to solve the following system of equations

$$\begin{cases} Y_0 = u_k + \Delta\tau \mathcal{F}(\tau_k)u_k, \\ Y_1 = Y_0 + \theta\Delta\tau \left( \mathcal{F}_1(\tau_{k+1})Y_1 - \mathcal{F}_1(\tau_k)u_k \right), \\ Y_2 = Y_x + \theta\Delta\tau \left( \mathcal{F}_2(\tau_{k+1})Y_2 - \mathcal{F}_2(\tau_k)u_k \right), \\ \tilde{Y}_0 = Y_0 + \frac{1}{2}\Delta\tau \left( \mathcal{F}(\tau_{k+1})Y_2 - \mathcal{F}(\tau_k)u_k \right), \\ \tilde{Y}_1 = \tilde{Y}_0 + \theta\Delta\tau \left( \mathcal{F}_1(\tau_{k+1})\tilde{Y}_1 - \mathcal{F}_1(\tau_k)u_k \right), \\ \tilde{Y}_2 = \tilde{Y}_x + \theta\Delta\tau \left( \mathcal{F}_2(\tau_{k+1})\tilde{Y}_2 - \mathcal{F}_2(\tau_k)u_k \right), \\ u_{k+1} = \tilde{Y}_2, \end{cases} \quad (30)$$

Guess	$\kappa_\nu$	$\mu_\nu$	$\sigma_\nu$	$\rho$
1	3.0	0.3	0.1	-0.2
2	5.0	0.6	0.2	-0.3
3	4.5	0.8	0.5	-0.15
4	2.0	0.4	0.45	-0.2
5	4.0	0.5	0.15	-0.35
6	3.5	0.35	0.5	-0.5

**Table 1:** Different initial guess sets for the initial coarse model calibration.

where we choose  $\theta = 3/4$  and implement  $u_k$  as a matrix [24]. The discrete boundary conditions in  $x$ -direction for a sufficiently small  $x_{\min}$  and a sufficiently large  $x_{\max}$  are set to  $V(x_{\min}, \nu, \tau) = \exp(-r\tau)$  and similarly  $V(x_{\max}, \nu, \tau) = 0$ . For the variance at  $\nu = \nu_{\min}$  and  $\nu = \nu_{\max}$  we use the ghost cell approach. Within this approach, additional grid points are computed at  $\nu_0 = \nu_{\min} - \Delta_\nu$  and  $\nu_{N_\nu+2} = \nu_{\max} + \Delta_\nu$  via zero-order extrapolation. For a detailed discussion, see [6, 17]. The integrals appearing in the gradient are computed by the trapezoidal rule.

## 5 Numerical Results

From the put options on the Nikkei 300 index on Dec. 31, 2012, we get one  $S_0$  and five different sets, each with the following parameters  $r$ ,  $q$ , and  $K$ . Thus, we specify  $M1, M2, M3, M4$  and  $M5$  for the different market data sets. Since the maturity must be small to use the European option price as a proxy for the Asian option price, we focus on  $T = 0.25$ . For the spatial discretization we set

$$\begin{aligned} x_{\max} = \log(1.2 * S_0), \quad N_x = 120, \quad \Delta_x = \frac{x_{\max}}{N_x} \quad \text{and} \quad x_{\min} = \Delta_x, \\ \nu_{\max} = 1, \quad N_\nu = 100, \quad \Delta_\nu = \frac{\nu_{\max}}{N_\nu} \quad \text{and} \quad \nu_{\min} = \Delta_\nu. \end{aligned} \quad (31)$$

and select  $\nu_0 = 0.05$ . As a result of this discretization, the strike price (and thus the kink in the payoff function) is approximated at a grid point. Therefore, we smooth the initial condition using the operator from Kreiss et al. [16] The time discretization uses

$$\tau_{\max} = T, \quad N_\tau = 170, \quad \Delta_\tau = \frac{T}{N_\tau} \quad \text{and} \quad \tau_{\min} = 0. \quad (32)$$

We consider six different initial guesses for the algorithm, see Table 1; each parameter set satisfies the Feller condition.

First, we focus on the gradient descent algorithm. For the algorithm, we set the iteration maximum to 51 and the terminal condition  $J_c(V, c(\xi_c)) < 10^{-3}$ . Since we are using a gradient-based algorithm, we can only expect to converge to a local minimum, to evaluate the descent over the iterations we use the relative reduction of the cost functional

$$r(\xi_c^0) = 100 \cdot \left( 1 - \frac{J(V(\xi_{\text{opt}}); V_{\text{data}})}{J(V(\xi_c^0); V_{\text{data}})} \right), \quad (33)$$

M \ Guess	1	2	3	4	5	6
1	99.97	99.91	99.98	99.66	99.99	<b>99.95</b>
2	99.97	99.92	99.98	99.81	99.98	99.96
3	99.97	<b>99.51</b>	<b>99.69</b>	99.50	99.98	<b>99.93</b>
4	99.97	<b>99.75</b>	<b>99.97</b>	99.72	99.96	<b>99.87</b>
5	<b>99.96</b>	99.92	99.98	99.92	<b>96.15</b>	<b>99.49</b>

**Table 2:** Cost functional reduction for the initial calibration of the coarse model.

M \ Guess	1	2	3	4	5	6
1	0.888	0.861	0.578	0.808	0.289	<b>1.058</b>
2	0.928	0.845	0.680	0.603	0.401	0.892
3	0.933	<b>4.638</b>	<b>9.914</b>	0.516	0.461	<b>1.702</b>
4	0.906	<b>1.639</b>	<b>1.248</b>	0.678	0.707	<b>3.542</b>
5	<b>1.031</b>	0.928	0.988	0.689	<b>56.160</b>	<b>17.323</b>

**Table 3:** Cost functional value for the optimal calibration of the coarse model, scaled by  $10^3$ .

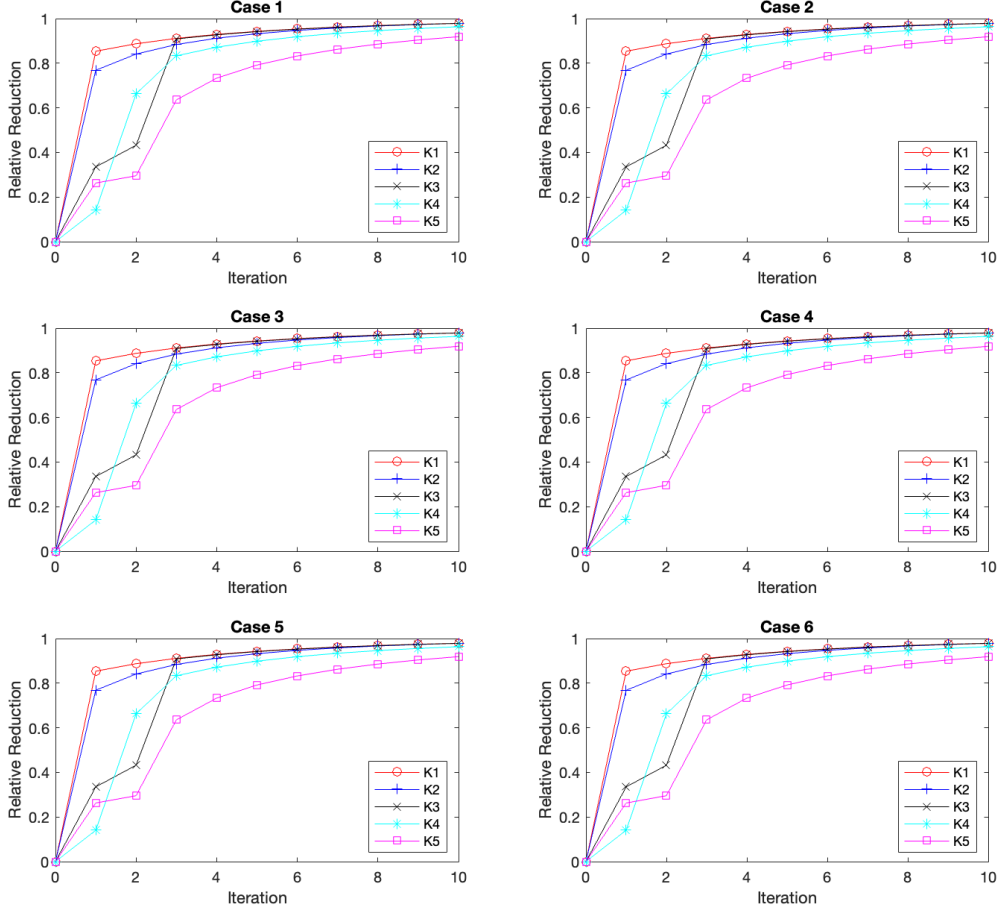
where  $\xi_c^0$  is the initial guess from Table 1.

Table 2 shows the cost functional reduction of the last and/or optimal cost function value. The bold values indicate instances where the maximum number of iterations was reached. We observe that the gradient descent algorithm calibrates the parameter  $\xi_c$  almost perfectly, even if we can't guarantee to find the global minimum. In Table 3, which shows the optimal value of the cost functional, we observe that for most of the test cases we reach the terminal condition of the gradient algorithm. The cases where the condition is not reached at the iteration maximum are shown in bold. The cases with the highest cost functional values correspond to the smallest cost function reduction. To illustrate the cost functional reduction per iteration, Figure 1 shows the value for the first 10 iterations. We observe that the maximum number of iterations can be significantly reduced depending on the desired accuracy.

These results show that the presented gradient descent algorithm is a viable choice for calibration. For a more detailed analysis of the gradient descent algorithm, see [5]. Within the space mapping, we limit ourselves to a maximum of four iterations, since the evaluation of the fine model is expensive. As a calibration measure we again use the relative cost functional reduction, the results are presented in Table 4. The results show that the space mapping approach has only slightly lower reduction rates, except for K5. One can improve this value by choosing a smaller  $\nu_0$ , as can be observed in K5a, where we choose  $\nu_0 = 0.03$ . Note that the choice of  $\nu_0$  is limited by the grid structure. In addition, the calibration to  $\nu_0$  can be included in the gradient descent algorithm.

Table 5 shows the overall cost functional reduction relative to the initial guess and Table 4 shows the optimal cost functional value. Similar to the gradient algorithm, we observe that when the cost functional reduction is small, the cost functional value is larger.

Since  $J(V_c(\xi_c^*); V_{\text{data}}) \approx J_f(V_f(\xi_f^0); V_{\text{data}})$  can be significantly worse than  $J(V_f(\xi_c^0); V_{\text{data}})$ , we present two figures. One figure shows the cost functional reduction per iteration, w.r.t.



**Figure 1:** Cost functional reduction for the first 10 Iterations for the initial calibration for the coarse model.

M \ Guess	1	2	3	4	5	6
1	98.57	98.75	99.07	96.73	99.52	99.89
2	99.66	99.21	99.96	72.41	98.93	99.22
3	98.01	99.64	99.98	98.33	91.67	99.39
4	92.42	99.20	93.42	98.44	98.05	99.74
5	28.22	53.71	64.48	52.74	42.48	71.84
5a	44.01	46.58	43.52	61.80	73.26	79.14

**Table 4:** Cost functional reduction for the calibration of the space mapping.

M \ Guess	1	2	3	4	5	6
1	0.021	0.006	0.012	0.007	0.005	0.001
2	0.004	0.002	0.001	0.003	0.009	0.009
3	0.012	0.001	0.022	0.011	0.015	0.0110
4	0.027	0.010	0.2010	0.024	0.014	0.007
5	1.159	1.124	1.480	1.290	1.110	1.046
5a	0.574	0.641	1.001	0.699	0.593	0.492

**Table 5:** Cost functional value for the optimal calibration of the space mapping.

M \ Guess	1	2	3	4	5	6
1	96.12	92.53	97.73	41.27	93.00	87.99
2	84.12	20.39	86.01	-1069.14	78.50	85.35
3	-28.13	-113.55	64.14	-5.97	-314.86	58.95
4	-317.71	-21.38	49.87	5.85	-100.91	43.96
5	-53.76	-1.80	39.41	10.54	-28.04	35.00

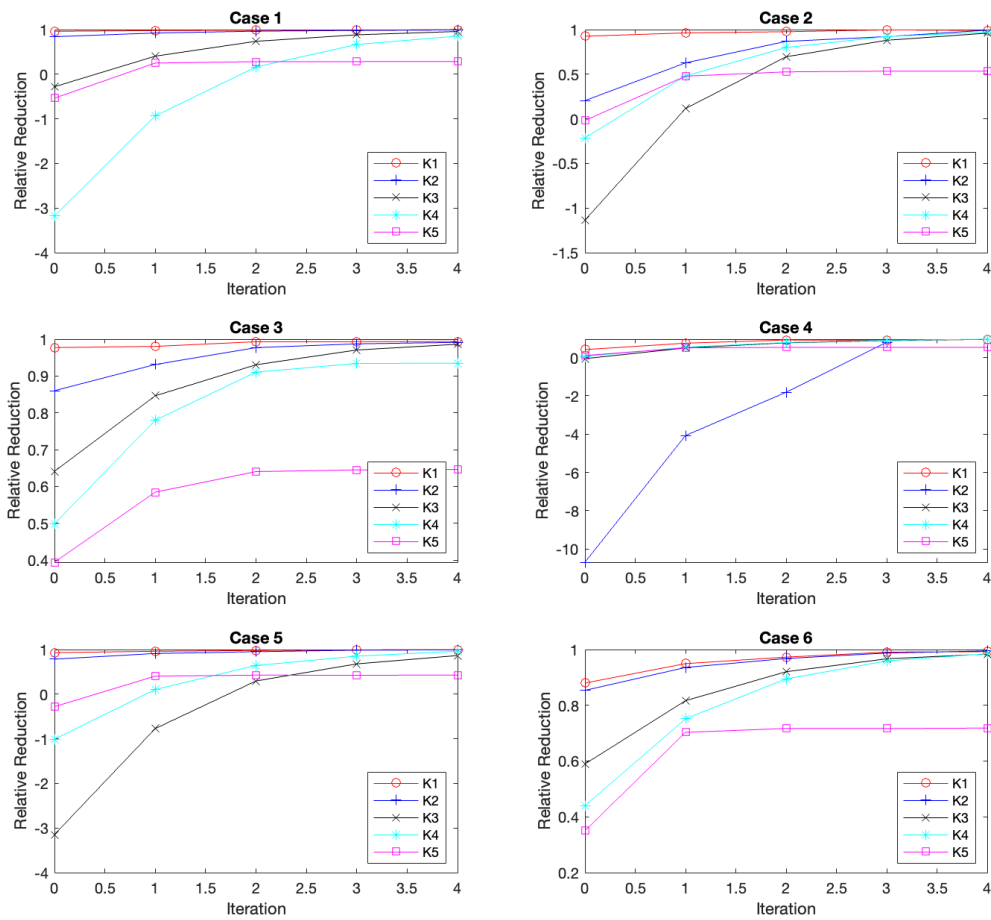
**Table 6:** Cost functional reduction for the initial guess and the calibration of the coarse model.

the initial guess  $\xi_c^0$ , see Figure 2, and the other one w.r.t. the optimal calibration parameters resulting from the initial coarse model calibration  $\xi_c^* = \xi_f^0$ , see 3. The Figures 2 and 3 and Table 6 show that even if  $\xi_f^0$  results in a higher cost functional value for the space mapping at iteration 0, the space mapping reduces the cost functional significantly, e.g. Guess 1 with  $M4$ , Guess 4 with  $M2$  as well as Guess 5 with  $M3$ .

Since the purpose of this article is to introduce space mapping to computational finance research as a proof-of-concept, no performance analysis is provided. However, by using faster or more accurate techniques for solving the Heston PDE and its adjoint equations, as well as for Heston calibration, the algorithm can be optimized.

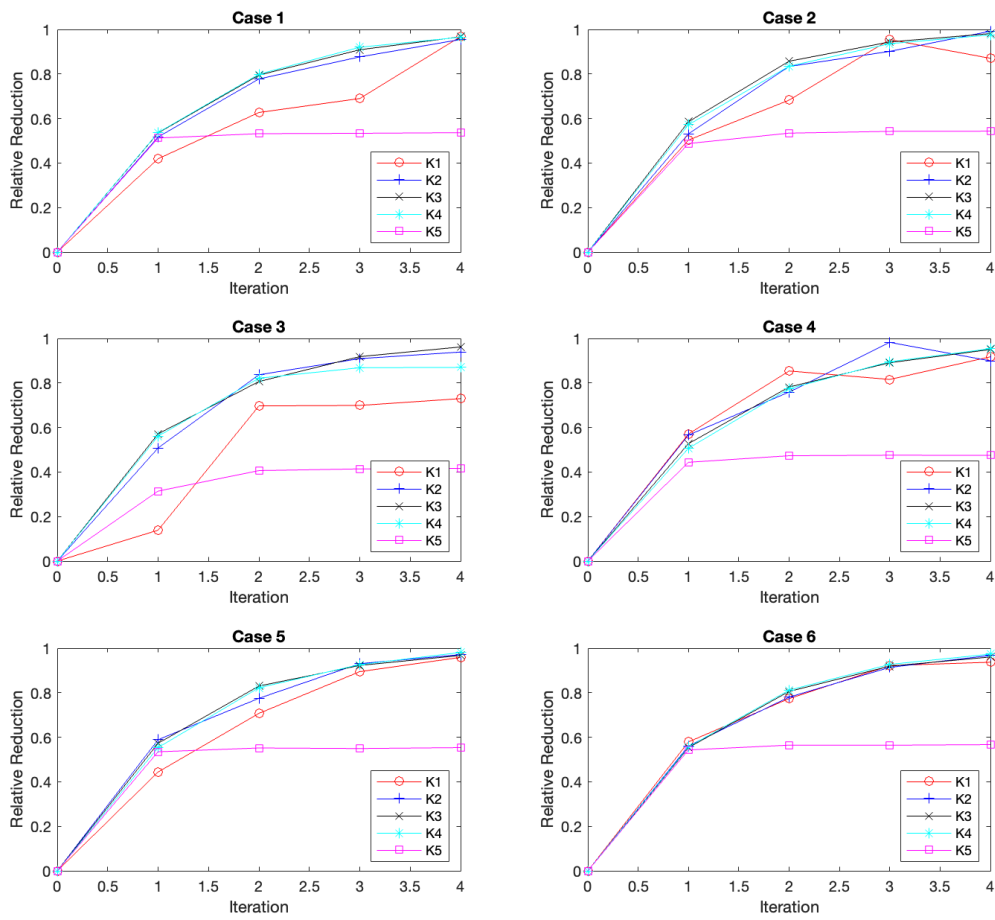
## 6 Conclusion and Outlook

The numerical results show the feasibility of the space mapping approach as a new calibration method in financial research. The numerical results are remarkable even in this proof-of-concept article and can be improved in several ways, e.g., by including time-dependent parameters, adding a calibration for  $\nu_0$ , and using improved computational methods for the coarse model optimization and for the fine model solver. In particular, the gradient descent algorithm can be easily adapted to account for time-dependent parameters to improve space mapping for pricing Asian options since they are time-dependent, see [5]. The space mapping approach can be applied to various other hierarchical problems in finance, e.g. model, temporal and spatial as well as option hierarchies. Therefore, this article is a first step towards integrating space mapping into financial applications.



**Figure 2:** Cost functional reduction within the space mapping algorithm for the fine model adjusted to  $\xi_{\text{init}}$ .





**Figure 3:** Cost functional reduction within the space mapping algorithm for the fine model adjusted to  $\xi_f^0$ .

## References

- [1] M. Bakr, J.W. Bandler, K. Madsen, and J. Søndergaard. Review of the space mapping approach to engineering optimization and modeling. *Optimization and Engineering*, 1(3):241–276, 2000.
- [2] J.W. Bandler, Q.S. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, and J. Søndergaard. Space mapping: the state of the art. *IEEE Trans. Microw. Theory Techn.*, 52(1):337–361, 2004.
- [3] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81(3):637–654, 1973.
- [4] Z. Bučková, M. Ehrhardt, and M. Günther. Fichera theory and its application in finance. In G. Russo, V. Capasso, G. Nicosia, and V. Romano, editors, *Progress in Industrial Mathematics at ECMI 2014*, pages 103–111, Cham, 2016. Springer.
- [5] A. Clevenhaus, C. Totzeck, and M. Ehrhardt. A gradient-based calibration method for the Heston model. *Int. J. Comput. Math.*, 101(9-10):1094–1112, 2024.
- [6] A. Clevenhaus, C. Totzeck, and M. Ehrhardt. A study of the effects of different boundary conditions on the variance of the Heston model. In K. Burnecki and M. Teuerle, editors, *Progress in Industrial Mathematics at ECMI 2023*. Springer, 2024.
- [7] Y. Cui, S. del Baño Rollin, and G. Germano. Full and fast calibration of the Heston stochastic volatility model. *Eur. J. Oper. Res.*, 263 (2):625–638, 2017.
- [8] D. Echeverría, D. Lahaye, and P.W. Hemker. Space mapping and defect correction. In *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, Berlin, Heidelberg, 2008.
- [9] A. Ferreiro-Ferreiro, J. García-Rodríguez, L. Souto, and C. Vázquez. A new calibration of the Heston stochastic local volatility model and its parallel implementation on gpus. *Math. Comput. Simul.*, 177:467–486, 2020.
- [10] M. Günther and A. Jüngel. *Finanzderivate mit MATLAB: Mathematische Modellierung und numerische Simulation*. Vieweg, 2 edition, 2010.
- [11] S.L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Fin. Stud.*, 6(2):327–343, 1993.
- [12] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, 2009.
- [13] B. Horvath, A. Muguruza, and M. Tomas. Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quant. Fin.*, 21(1):11–27, 2021.

- [14] W. Hundsdorfer. Accuracy and stability of splitting with stabilizing corrections. *Appl. Numer. Math.*, 42(1-3):213–233, 2002.
- [15] D. Khalife, J. Yammine, S. Rahal, and S. Freiha. Pricing Asian and barrier options using a combined Heston model and Monte Carlo simulation approach with artificial intelligence. *Math. Modell. Engrg. Probl.*, pages 1690–1698, 2023.
- [16] H.-O. Kreiss, V Thomée, and O. B. Widlund. Smoothing of initial data and rates of convergence for parabolic difference equations. *Commun. Pure Appl. Math.*, 23:241–259, 1970.
- [17] P. Kútik and K. Mikula. Diamond-cell finite volume scheme for the Heston model. (*DCDS-S*, 8(5):913–931, 2015.
- [18] I. Leite, J. Yamim, and L. da Fonseca. The DeepONets for finance: An approach to calibrate the Heston model. In G. Marreiros et al., editors, *Progress in Artificial Intelligence*, pages 351–362, Cham, 2021. Springer.
- [19] S. Liu, A. Borovykh, L.A. Grzelak, and C.W. Oosterlee. A neural network-based framework for financial model calibration. *J. Math. Ind.*, 9:1–28, 2019.
- [20] S. Liu, C.W. Oosterlee, and S. M. Bohte. Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):16, 2019.
- [21] F. Mehrdoust, I. Noorani, and A. Hamdi. Two-factor Heston model equipped with regime-switching: American option pricing and model calibration by Levenberg-Marquardt optimization algorithm. *Math. Comput. Simul.*, 204:660–678, 2023.
- [22] S.S. Mikhailov and U. Nögel. Heston’s stochastic volatility model implementation, calibration and some extensions. In *Willmot Magazine*. John Wiley and Sons, 2004.
- [23] R. Pinnau N. Marheineke and E. Reséndiz. Space mapping-focused control techniques for particle dispersions in fluids. *Optim. Engrg.*, 13:101–120, 2012.
- [24] L. Teng and A. Clevenhuis. Accelerated implementation of the ADI schemes for the Heston model with stochastic correlation. *J. Comput. Sci.*, 36:101022, 2019.
- [25] L. Teng, M. Ehrhardt, and M. Günther. On the Heston model with stochastic correlation. *Int. J. Theor. Appl. Fin.*, 6, 2016.
- [26] C. Totzeck and R. Pinnau. Space mapping-based receding horizon control for stochastic interacting particle systems: dogs herding sheep. *J. Math. Ind.*, 10:1–19, 2019.
- [27] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen*. Springer, 2009.