PARAMETER IDENTIFICATION FOR UNDERDETERMINED SYSTEMS ARISING IN OPTION PRICING MODELS AND NEURAL NETWORKS

Dissertation

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

Dem Fachbereich IV der Universität Trier
vorgelegt von

MICHAELA SCHULZE

Trier 2002

Eingereicht am 5. Juni 2002

1. Gutachter: Prof. Dr. E. W. Sachs
2. Gutachter: Prof. Dr. P. L. Toint

Tag der mündlichen Prüfung: 12. August 2002

Acknowledgements

I would like to offer my thanks to the many people who have contributed to the success of this dissertation and to make my mathematical live over the last years a rewarding experience. Without the promotion and support of teachers, colleagues, friends and relatives, the completion of this work would not have been successful. I am indebted to this people for their assistance, inspiration and encouragement.

Especially, I would like to thank my advisor Prof. Dr. Sachs who provided a substantial contribution by advising, encouraging and supporting this work in many ways. Our numerous discussions as well as his very helpful comments always have motivated me to keep up in writing this dissertation.

Furthermore, I am grateful to my coreader Prof. Dr. Toint from the Facultés Universitaires Notre Dame de la Paix in Namur, Belgium, for his interest in my work and for the valuable comments and discussions.

Thanks are extended to my colleagues for many lively and open discussions as well as to my friends for their understanding and many cheerful and happy hours. In particular, I would like to express my gratitude to Jerri Lynn Sayers who additionally helped me proofreading the manuscript as well as to Wolfram Himpel for supporting me over a long period of this work.

Finally, I would like to thank my parents and grandparents for their constantly moral as well as financial support of my complete academic career.

Contents

1 **Introduction** **1**

2 **An Iterative Inexact Gauss–Newton Method** **13**

- 2.1 The Nonlinear Least–Squares Problem 14
- 2.2 The Inexact Gauss–Newton Method 16
  - 2.2.1 The Gauss–Newton Method 16
  - 2.2.2 Globalization of the Algorithm: Introduction of a Trust–Region Strategy . . 18
  - 2.2.3 Iterative Solution of the Trust–Region Subproblem 21

3 **Convergence** **29**

- 3.1 Global Convergence of the Inexact Gauss–Newton Method 29
- 3.2 Local Convergence Analysis of the Inexact Gauss–Newton Method 32
  - 3.2.1 Known Local Convergence Results for the Gauss–Newton Method 32
  - 3.2.2 Difficulties in the Convergence Analysis of the Inexact Gauss–Newton Method Applied to Underdetermined Problems 34
  - 3.2.3 Local q–Convergence Analysis of an Arbitrary Inexact Gauss–Newton Iteration 35
  - 3.2.4 Local q–Convergence Analysis of the Inexact Gauss–Newton Iteration Generated by the Proposed Algorithm 39
  - 3.2.5 Local q–Convergence Analysis of the Inexact Gauss–Newton Iteration for Overdetermined Problems 39
  - 3.2.6 Local q–Convergence Analysis of a Specific Inexact Gauss–Newton Iteration 41

4 **Forecasting with Neural Networks** **43**

- 4.1 Description of Neural Networks 43
- 4.2 Evaluation of the Jacobian 48
- 4.3 Sensitivity Analysis 54
- 4.4 Numerical Results 55
  - 4.4.1 Convergence Rates 56
  - 4.4.2 Neural Networks in Forecasting Stock and Index Prices 58
  - 4.4.3 Preparation of the Neural Network and the Training Patterns 60
  - 4.4.4 Training of Large–Scale Networks 64

4.4.5 Forecasting and Performance 68
4.4.6 Sensitivities 73
4.4.7 Concluding Remarks on Neural Networks in Financial Forecasting 76

5 Constructing the Instantaneous Volatility Function 77
5.1 Motivation 78
5.2 An Extended Black–Scholes Model for European Calls 82
5.2.1 The “Dual” European Call Price Model 82
5.2.2 Transformation of the “Dual” Extended Black–Scholes Equation 85
5.2.3 Existence and Uniqueness of Solutions to the Transformed “Dual” Extended Black–Scholes Equation 86
5.3 Determination of the Instantaneous Volatility Function 90
5.4 Determination of the Discrete Instantaneous Volatility Function 91
5.4.1 Derivation of Boundary Conditions 92
5.4.2 Discretization Scheme for the Transformed Extended Black–Scholes Model 94
5.4.3 Convergence of the Discretization Scheme 104
5.4.4 Determination of the Discrete Instantaneous Volatility Function 123
5.5 Evaluation of the Jacobian 133
5.6 Numerical Results 144
5.6.1 Synthetic European Call Option Examples 145
5.6.2 Construction of the Instantaneous Volatility Function for the German Stock Index DAX 154

6 Conclusions 159

A The Parabolic Initial–Boundary–Value Problem 163
A.1 Existence and Uniqueness of the Solution 163

B Consistency of the Finite Difference Scheme 167
B.1 Proof of Theorem 5.4.10 167

Bibliography 173

Chapter 1

Introduction

In this thesis, we study the convergence behavior of an efficient optimization method used for the identification of parameters for underdetermined systems.

The research in this thesis is motivated by optimization problems arising from the estimation of parameters in finance. In the following, we are especially concerned with the valuation of European call options as well as neural networks used to forecasting stock market indices.

Assuming that an asset price satisfies the stochastic differential equation

$$dS = \mu(S,t)Sdt + \sigma(S,t)SdW_t$$

with $\mu, \sigma : I\!\!R^+ \times [0, \bar{T}] \to I\!\!R$ are the drift and volatility term, respectively, and W_t is a Brownian motion, then the value of a European call option $C(K,T)$ in dependence on the strike price K and the maturity T satisfies the parabolic differential equation

$$C_T - \tfrac{1}{2}\sigma(K,T)^2K^2C_{KK} + (r(T) - d(T))KC_K + d(T)C = 0$$

subject to the initial condition

$$C(\sigma(\cdot,\cdot); K, 0) = \max(S - K, 0)\,,$$

see Andersen and Brotherton–Ratcliffe [1] and Dupire [32]. The valuation of the European call option according to this model requires information about the volatility parameter $\sigma(K,T)$ which is not explicitly observable in the market. For small maturities T and strike prices K that are close to the current asset price, usually, the volatility $\sigma(K,T)$ is computed implicitly from market European call options. However, for the valuation of European call options with long maturities or that are far out or in the money, this procedure does not work. In accordance with the dependence of the European call price on the volatility parameter $\sigma(K,T)$, we will slightly change the notation for this call value to $C(\sigma(\cdot,\cdot); K, T)$. For the numerical solution of the problem, we discretize the parabolic differential equation, i.e. the infinite dimensional functions are replaced by finite dimensional approximations. In the following, let $\bar{\sigma}$ and $z(\bar{\sigma})$ be the finite dimensional approximations of the volatility parameter and the call value, respectively. Then, we will estimate the finite dimensional version of the unknown volatility $\bar{\sigma}$ by using market European call prices, i.e. $\bar{\sigma}$ will be adapted such that the model European call prices approximate the corresponding counterparts in the market.

Moreover, we will examine an application of neural networks to forecasting stock market indices. The nonlinear input–output map of a neural network can be described by a function $y^l(w;t) = s$ where t is the input, s is the output generated by the neural network and w is the vector of weights and threshold values which are responsible for the adjustment of the neural network to a particular problem. The identification of this parameter vector w, which is also called the training of the neural network, is carried out through a set of appropriate examples representing the particular application.

Hence, both models can be described by a function $y(x,t) = s$ where s represents the output of the particular model which in each case depends nonlinearly on n unknown parameters $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ and the input data t. The unknown parameter vector x will be determined by using some given observations. If (t_i, s_i), $i = 1, \ldots, p$, are the observed input–output patterns and

$$r_i(x) = y(x, t_i) - s_i\,, \quad i = 1, \ldots, p\,,$$

is the discrepancy between the model evaluated at the parameter vector x and the observations then the identification of the parameters in each of these models can be posed as the nonlinear least–squares problem

$$\text{Minimize} \quad \tfrac{1}{2}\, \|R(x)\|_2^2$$

where $R : \mathbb{R}^n \to \mathbb{R}^m$ is defined by $R(x) = (r_1(x), \ldots, r_p(x))^T$ and $\|\cdot\|_2$ is the Euclidean norm.

In neural networks, the dimension n depends on the number of neurons in each layer of the network and on the number of layers. In general, this number tends to grow very quickly when adding additional neurons, e.g. in the input layer. In the case that neural networks are used to forecasting stock market indices, the amount of input data is very large since the behavior of stock market indices is usually influenced by a variety of factors. Hence, many different time series can be used to provide input data for the neural network such that n becomes very large in this case. The dimension m of the range space of R depends on the number of neurons in the output layer of the neural network and the number of training patterns (t_i, s_i), $i = 1, \ldots, p$. When sufficient data are available, in particular in cases like time series, then this dimension also increases quickly, however, not as fast as n gets large. Hence, in this case usually $n \gg m$ such that the nonlinear least–squares problem is underdetermined.

This relationship also holds for the nonlinear least–squares problem resulting from the determination of the discretized volatility function $\bar{\sigma}$ in the European call price model. In this thesis, we examine a model for the valuation of European call options where the volatility parameter $\sigma(K,T)$ of the underlying asset is a real valued function defined on the rectangular domain $\mathbb{R}^+ \times [0, \bar{T}]$. Hence, the dimension n of the discrete version $\bar{\sigma}$ of the volatility parameter depends on the discretization size of the grid approximating $\mathbb{R}^+ \times [0, \bar{T}]$ such that in this problem n tends to grow very quickly, too. The dimension m of the range space of the residual function R depends on the number of market European call prices with the same underlying. However, option prices are only quoted on an exchange for exercise prices at discrete intervals [107] and which are close to the current value of the underlying, i.e. at–the–market options. Furthermore, on an exchange options are only traded for a small number of maturities. Therefore, the dimension m usually is much smaller than the dimension n of the parameter space, i.e. $n \gg m$.

In Chapter 2, we are concerned with the solution of underdetermined nonlinear least–squares problems. For this class of problems, it is likely that their residual vector $R(x)$ at the solution is fairly small. In this case, the Gauss–Newton method is a promising optimization method for the solution of nonlinear least–squares problems [15], [25]. The Gauss–Newton method approximates in each step the nonlinear residual vector $R(x)$ by an affine model $m^a(\delta x) = R(x) + J(x)\,\delta x$ such that in each iteration the linear least–squares problem

$$\text{Minimize} \quad \tfrac{1}{2}\, \|J(x)\,\delta x + R(x)\|_2^2 \,,$$

or, equivalently, the linear system of equations

$$J(x)^T J(x)\,\delta x = -J(x)^T R(x)$$

has to be solved where $J(x) \in I\!R^{m\times n}$ is the Jacobian of the residual vector $R(x)$. As mentioned above, we are concerned with problems where the dimension of the vector $x \in I\!R^n$ is large.

Unfortunately, $J(x)$ is not only large but also dense, so that it is impossible to store the matrix $J(x)$ or even $J(x)^T J(x)$. Hence, we are not able to use direct methods for the solution of linear systems in this case. However, in both models, we can evaluate the Jacobian applied to a vector at a reasonable cost, i.e. $J(x)\,\delta v$ or $J(x)^T \delta v$ is available, see for instance Sachs, Schulze [91] for the evaluation of the Jacobian resulting form the neural network model. The structure of these routines will be close to the ideas of automatic differentiation where the evaluation of the matrix–vector product $J(x)\,\delta v$ respective $J(x)^T \delta v$ is much cheaper than the computation of $J(x)$ itself, see for instance Griewank [46], Saarinen, Bramley and Cybenko [90]. Therefore, we are able to use an iterative scheme to solve the linear least–squares subproblem in every iteration of the Gauss–Newton method [7].

An overview of iterative methods for the solution of linear systems is given in [36] and for the solution especially of linear least–squares problems can be found in [7] and the references cited therein. The coefficient matrix $J(x)^T J(x)$ in the normal equation is symmetric. Thus, we will use a method which exploits this structure like the conjugate gradient method, which is due to Hestenes and Stiefel [56]. This method, that belongs to the Krylov subspace methods, computes in each step an optimal approximation of the solution to the normal equation in the nested set of Krylov subspaces. For the computation of these iterates, it is necessary to evaluate the matrix–vector products $J(x)\,\delta v$ and $J(x)^T \delta v$ and to perform some additional $3n + 2m$ multiplications. The storage requirements are held constant over the process. Faber and Manteuffel showed that, except for a few anomalies, there do not exist Krylov subspace methods for more general linear systems that fulfil this optimality condition by keeping the work and storage requirements in each of the iteration low and roughly constant at the same time [34], [35]. For example, in the GMRES method, a Krylov subspace method for nonsymmetric linear systems, it is necessary to store an orthogonal basis of the particular Krylov spaces which, in our case, can lead to storage problems since the vector x of the unknowns is very large. Moreover, the vectors in the basis can become nonorthogonal, especially when $J(x)$ is ill–conditioned or many iterations are needed [64] which may lead to inaccurate approximate solutions.

In this thesis, we will use the special version of the conjugate gradient method for the solution of linear least–squares systems which is already presented in the original paper on conjugate gradients by Hestenes and Stiefel [56]. In [80], this algorithm is denoted CGLS and in [36] it is called CGNR. It is also possible to apply the version of the conjugate gradient method for general symmetric positive definite systems to the normal equations which would result in an explicit formation of the matrix–vector product $J(x)^T J(x)\,\delta v$. However, the formation of the matrix–vector product $J(x)^T J(x)\,\delta v$ can lead to poor performance especially when the condition number $\kappa(J(x))$ of $J(x)$ is large [8], i.e. when the system is ill–conditioned, since a small perturbation in $J(x)^T J(x)$, e.g. by roundoff, may change the solution much more than perturbations of similar size in $J(x)$ itself [7]. Furthermore, Björck shows in [7, Section 1.4] that in the case of a small residual at the solution of the linear least–squares problem the condition number for the linear least–squares problem will be smaller than the one for the normal equations, i.e. methods for the linear least–squares problem usually will be more reliable than normal equations methods [18].

Moreover, note that the version of the conjugate gradient method for the solution of the linear least–squares problem is equivalent to the LSQR method due to Paige and Saunders [80], a stable method [8] based on the bidiagonalization procedure of Golub and Kahan [42].

An additional difficulty with the underdetermined linear least–squares problem is the fact that its solution cannot be expected to be unique. However, if the initial approximation of the solution in the conjugate gradient method is chosen properly, then the conjugate gradient method computes the unique solution to the linear least–squares problem with minimal norm [54], [55], i.e. in this case the iterates of the conjugate gradient method converge to $\delta x_* = -J(x)^+ R(x)$ with $J(x)^+$ is the Moore–Penrose inverse or pseudoinverse of $J(x)$. This is the solution of the Gauss–Newton subproblems which is desirable in the problems resulting from the formulation of the training process in neural networks and the determination of the volatility function $\bar{\sigma}$ in the European call price model.

In neural networks, the decision about sending a signal from one neuron to another depends on the evaluation of a sigmoidal function [109], e.g. the logistic function $\sigma(r) = 1/(1 + e^{-2r})$ which we will use in the model described in Chapter 4 of this thesis. For large arguments, i.e. $|r|$ is large, this function is almost constant. Hence, the residual function for the training of the neural network, a composition of this activation function, does not change much if large weights are present such that those are inefficient for the training of the neural network. Thus, in neural networks the solution to the linear least–squares problem with minimal norm is desirable, since a small norm of the variable means that each of the components of the variable, i.e. the weights in this case, are small, too.

In the European call price model, the volatility function $\sigma(K, T)$ is supposed to be a sufficiently smooth function. This smooth behavior of the volatility function is modelled in the discrete problem formulation by restricting the size of the step $\delta\bar{\sigma}$ of the discrete version of the volatility function $\bar{\sigma}$ according to a weighted norm, i.e. $\|\delta\bar{\sigma}\|_{\bar{D}} = \sqrt{\langle \delta\bar{\sigma}, \bar{D}\,\delta\bar{\sigma}\rangle}$ with $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. The weight matrix $\bar{D}$ will be a slight modification of the discrete version of the differentiation operator (note that we cannot use the exact discrete version of the differentiation operator D because

D has a nontrivial null space). Therefore, we want to generate a sequence of conjugate gradient iterates which converge to a minimal $\bar{D}$–norm solution. Since the matrix $\bar{D}$ is symmetric positive definite, the square root $\bar{D}^{1/2}$ exists and

$$\|\delta\bar{\sigma}\|_{\bar{D}}^2 = \langle \delta\bar{\sigma}, \bar{D}\,\delta\bar{\sigma}\rangle = \langle \bar{D}^{1/2}\delta\bar{\sigma}, \bar{D}^{1/2}\delta\bar{\sigma}\rangle = \|\bar{D}^{1/2}\delta\bar{\sigma}\|_2^2\,.$$

Hence, a transformation of the variables $\delta y = \bar{D}^{1/2}\delta\bar{\sigma}$ leads to the Gauss–Newton subproblems

$$\text{Minimize} \quad \tfrac{1}{2}\,\|(J(\bar{\sigma})\,\bar{D}^{-1/2})\,\delta y + R(\bar{\sigma})\|_2^2\,.$$

These transformed problems have the same structure than those when a preconditioner is introduced into the conjugate gradient method. Thus, in the Gauss–Newton algorithm for the solution of the nonlinear least–squares problem resulting from the determination of the discrete volatility function $\bar{\sigma}$ we will solve the linear least–squares subproblems with the preconditioned version of the conjugate gradient method. The weight matrix $\bar{D}^{1/2}$ will take the role of the preconditioner. However, note, that $\bar{D}^{1/2}$ is not a preconditioner in the original sense which is introduced to accelerate the convergence of the algorithm but is established because of the given facts of the underlying problem. Furthermore, $\bar{D}^{1/2}$ will be constant for the complete iteration process such that a factorization of $\bar{D}^{1/2}$ has to be computed only once at the beginning of the algorithm.

As we will see in the convergence analysis of the algorithm, the sequence of iterates generated by the pure Gauss–Newton method will converge to a solution only when the initial iterate of the algorithm is already a sufficiently “good” approximation of the solution. A larger convergence region can be achieved by introducing a globalization technique such as a line search technique or a trust–region strategy.

For the solution of the training problem in neural networks, a line search technique is used in the Backpropagation algorithm; the traditional method for the training of neural networks that uses steepest descent information. This means that the convergence of this method to a stationary point, starting at an arbitrary starting point, is showed by introducing a certain step length λ [71], [47] which, in the neural network context, is called the learning rate. Using a line search technique in the Gauss–Newton method has the disadvantage that in case of a rank deficient Jacobian these techniques always must include some estimation strategy for the rank of the Jacobian to compute a reliable Gauss–Newton direction.

Hence, in this thesis we use a trust region strategy to implement a globalization strategy since this strategy enforces a regularization of the Gauss–Newton subproblems when the Gauss–Newton direction has bad descent properties. Thus, in every iteration of the Gauss–Newton method we solve the restricted linear least–squares problems

$$\text{Minimize} \quad \tfrac{1}{2}\,\|J(x)\,\delta x + R(x)\|_2^2\,, \qquad \text{s.t.} \qquad \|\delta x\|_{\bar{D}} \le \Delta\,,$$

see for instance Heinkenschloß [52]. Using the trust region strategy has the advantage that it can be combined with the conjugate gradient solver [105], [98], i.e. the conjugate gradient method can be used for the solution of the trust–region subproblems in each iteration of the Gauss–Newton method. The resulting algorithm is closely related to the Levenberg–Marquardt method [69], [72].

The global solution of the Gauss–Newton–Trust–Region subproblems is also the solution of the Levenberg–Marquardt subproblems for a special λ. The difficulty in the Levenberg–Marquardt method consists just in the choice of this parameter λ. Moré presented in [75] an implementation of the Levenberg–Marquardt method in a trust–region framework. The advantage of the trust–region method in comparison with the Levenberg–Marquardt method is the self adapting procedure for the size of the trust–region and, thus, for the size of the steps.

The requirements on the presented method are determined by the characteristics of the two model problems; however, the results in the following two chapters in this thesis will be in a fairly general setting to make the method applicable to a wide class of large scale underdetermined nonlinear least–squares problems. An important aspect for the efficient implementation of the proposed algorithm in the case of large scale problems is the performance of the matrix–vector products $J(x)\delta v$ and $J(x)^T\delta v$. Later on, we will show how to make the training of neural networks and the determination of the volatility function amenable to large scale optimization methods by exploiting the particular problem structure inherited from each of the models and by providing subroutines for an efficient evaluation of $J(x)\delta v$ and $J(x)^T\delta v$.

As mentioned above, the global convergence of the presented algorithm follows from the introduction of the trust–region strategy. The global convergence theory of the latter to a stationary point is well established, see for example [82], [83], [84], [102], [76], [18]. A significant aspect in the proof of the global convergence of trust–region methods is a sufficient model decrease in each step of the algorithm. Steihaug shows in [98] that the iterates generated by the modified conjugate gradient method satisfy this sufficient model decrease condition. At the beginning of Chapter 3, we will outline the main ideas of the proof of the global convergence of the algorithm presented in Chapter 2.

Usually, nonlinear least–squares problems are examined for overdetermined problems, i.e. problems where the dimension of the range space of the residual vector R is larger than the dimension of the parameter vector x. Thus, there are many papers on the convergence behavior of Gauss–Newton methods applied to this class of problems; see for instance Deuflhard and Heindl [30], Deuflhard and Apostolescu [29], Dennis and Schnabel [25], Schaback [93], and for the inexact Gauss–Newton method, Dennis and Steihaug [26]. However, the area of underdetermined least–squares problems has been studied less actively. Boggs shows in [11] the local convergence of the Ben–Israel iteration combined with a step size control to a solution in the general case, i.e. for overdetermined as well as for underdetermined problems. Furthermore, Bock [10] proves the r–linear convergence of the exact Gauss–Newton method using the minimal norm solution of the linear least–squares problems.

In Chapter 3, we analyze the local convergence properties of the inexact Gauss–Newton method for a more general class of residual vectors R. Since in general the Jacobian $J(x)$ has a nontrivial null space, the convergence rate of the iterates has to be considered in the orthogonal subspace of the null space of $J(x)$. When in each Gauss–Newton iteration an arbitrary solution of the Gauss–Newton subproblem is chosen for the correction of the Gauss–Newton iterate, then we show that this sequence converges q–linearly to a solution on the orthogonal complement of the null space of $J(x)$. This result holds also when the solution is computed inexactly. If the residual function

R is zero at the solution, then this sequence converges even q–quadratically to a solution on the above mentioned subspace. Thus, these results also hold for the sequence generated by the above described inexact Gauss–Newton method that computes a truncated minimal norm solution to the linear Gauss–Newton subproblems in each iteration. Furthermore, we show that in the special case of a overdetermined residual function R with a full rank Jacobian these results reduce to the known results in the literature. Finally, for a particular sequence of Gauss–Newton iterates which is generated with a specific solution of the Gauss–Newton subproblems the q–linear convergence of the iterates to a local solution of the nonlinear least–squares problem can be shown without restriction to a specific subspace. However, this sequence can only chosen theoretically since information about a minimizer of the nonlinear least–squares problem are necessary to construct this Gauss–Newton sequence.

In the following chapters, we apply the proposed inexact Gauss–Newton method to two underdetermined nonlinear least–squares problems arising in finance.

In the first application, we are concerned with neural networks applied in forecasting economic data such as stock or stock index values. The question about the future development of the stock market is equally interesting for experts as well as theorists. Because of the special economical importance of the knowledge of the future market development, it is not surprising that a variety of theories with corresponding analysis instruments were developed [87].

The stock market is characterized by its highly complex structure whose co–operation of the variables is neither theoretically uniquely explainable nor empirically exactly describable [5]. Moreover, there exists a large variety of variables in the stock market which are not all obvious and which change over time. Furthermore, there is evidence that also the interactions of these variables change and that they are nonlinear [81]. Since neural networks are qualified to simultaneously model a variety of influencing factors and their nonlinear interactions, they offer a promising mathematical instrument in forecasting. The advantage of the application of neural networks in forecasting is that the underlying function form which generates the data does not need to be known in advance. The neural network rather tries to find this function during the learning process [65].

However, we will see that complete freedom in the choice of the model is also not given if neural networks are used. An important part in the determination of a first idea of the model plays the choice of the data since the data set should be sufficiently large and representative concerning the task of stock or index price forecasting so that the structure of the formation of prices can be marked. Thus, the experts opinion regarding the explanation of the formation of prices and the valuation approach for the calculation of prices will play an important role in the choice of the data. The two main existing model classes in this context are the fundamental analysis and the technical analysis [16]. The advantage of neural networks is that they can be assigned either to the technical or the fundamental approach by feeding exclusively either technical oriented or fundamental oriented input data into the neural network or they combine these two approaches by simultaneously analyzing both variables which are assigned to the technical analysis and variables which are assigned to the fundamental analysis in solely one computation.

As mentioned above, the training process of the neural network, i.e. the determination of the unknown weights and bias in the neural network, can be described by an underdetermined nonlinear least–squares problem such that the proposed inexact Gauss–Newton method seems to be a promising algorithms for its solution. If the training process is carried out until the error of the outputs of the neural network and the target outputs in the training patterns is minimized, then the neural network strives to perfectly map the presented data. However, the extracted structure can not be used for a generalization since time series not only contain deterministic behavior but also a certain portion of noise that should not be modelled by the network. To avoid this overfitting behavior, we stop the inexact Gauss–Newton method by using the Stopped–Training method, a method that determines the time during the learning process when the ability of generalization of the network decreases.

A comparison of the training results of the proposed inexact Gauss–Newton method and the backpropagation algorithm will show the efficiency of the former method. We will see that using the inexact Gauss–Newton method for the solution of the nonlinear least–squares problem decreases the computation time of the training process extremely. Moreover, the inexact Gauss–Newton method enables the treatment of very large neural networks, even with half a million variables. This opens a new area of application for neural networks. In addition, we give numerical results to illustrate the relevance of the local convergence theorems. In particular, one can see that the projection of iterates is needed to analyze the rate of convergence properly.

Furthermore, we examine the quality of the forecasts generated by the neural network. This will be done by comparing the output computed by the network with the target output and by computing the rate of right trends and the trading profit of a given trading strategy for the neural network. We obtain some very promising results for applying neural networks to forecasting stock and index values. However, these results also elucidate the time–consuming process of finding an optimal network.

One disadvantage mentioned in connection with neural networks is that the knowledge found during the training process is difficult to extract since it is distributed among the different weights and biases of the neural network. We will show that it is possible to determine the sensitivities of the optimal output of the neural network with regard to small variations of the various components of the input vector. At the end of this chapter, we will compute these sensitivities for different neural networks.

Finally, in the last chapter of this thesis, we are concerned with the valuation of European call options. A well known model in this context is the Black–Scholes model [9] which assumes that the underlying asset follows an one–factor diffusion process with constant drift and volatility parameters and that the known risk–free interest rate and the dividend yield are constant, too. Using arbitrage–free pricing techniques, it follows that in this case the European call price function is described by a parabolic initial–value problem with constant coefficients. Moreover, since the parabolic differential equation can be transformed to the one–dimensional heat–equation, there exists also an explicit representation of the value of the European call option in the Black–Scholes model, the so–called Black–Scholes formula. The only not directly in the market observable parameter in this formula is the volatility which can be either estimated empirically from historical data or numerically by inverting the Black–Scholes formula to compute the volatility implied by an option price observed in the market. Despite the widespread use of the Black–Scholes model in financial practice, it is generally realized that the assumptions of the model are not all conforming to reality. The existence of term structures in interest rates and dividends, for instance, indicates that these two parameters are at least functions dependent on time. Furthermore, Rubinstein [88] empirically showed that the Black–Scholes formula became increasingly unreliable over time and that the implied volatility of market European call options varies for options with different strike prices. Moreover, Dupire [32] exhibited a variation of the volatility also with respect to the maturity. The existence of this so–called volatility smile or volatility skew indicates that the price process of the underlying asset does not correspond to an one–factor diffusion process with constant drift and volatility parameter [1].

In financial practice, these deficiencies are overcome by managing tables of values for the interest rate, the dividend yield and the volatility parameter belonging to different option maturities and strike prices. Unfortunately, this approach works only for European call options with short maturities and strike prices that are close to the current value of the underlying. However, financial institutions also have an interest in volatility values that belong to options with long maturities or strike prices that are not close to the current asset value since they might deal with individual inquiries or because they might want to hedge older positions in their portfolio that were held speculative so far. Moreover, this approach does not work for pricing of more complicated options such as exotic options or options with early exercise features [1].

There exists a variety of approaches trying to be consistent with the volatility smile. Some of these approaches, see for instance Merton [74], Hull and White [59], include an additional source of risk which destroys the completeness of the model. However, Derman and Kani [27], Dupire [32] and Rubinstein [88] have independently shown that an one–factor diffusion model for the asset value in which the drift and the volatility term are deterministic functions dependent on the asset value and the time is sufficiently rich to be consistent to most reasonable volatility smiles [67]. The advantage of this model is that it maintains the completeness. Moreover, Dupire [32] has shown that this volatility function can be extracted from market European call options when they are available for all conceivable strike prices and maturities.

Since this one–factor diffusion model maintains completeness, arbitrage–free pricing techniques can be used to show that the European call price function $C(\sigma(\cdot,\cdot);S,t)$ satisfies the following deterministic parabolic initial–value problem with nonconstant coefficients and with one of the coefficients consisting of the deterministic volatility function:

$$\begin{aligned} &C_t \;+\; \tfrac{1}{2}\,\sigma^2(S,t)\,S^2\,C_{SS} \;+\; (r(t)-d(t))\,S\,C_S \;=\; r(t)\,C\,, && (S,t)\in I\!R^+\times(0\,,T] \\ &C(\sigma(\cdot,\cdot);S,T) \;=\; \max(S-K,0)\,, && S\in I\!R^+\,. \end{aligned}$$

In the following we will call this model the extended Black–Scholes model because of its similarity to the original Black–Scholes model. Moreover, Dupire [32] and Andersen and Brotherton-Ratcliffe [1] have derived an additional model for European call options also consisting of a deterministic parabolic initial–value problem by using the Feynman–Kac Theorem and the Kolmogorov forward equation:

$$\begin{aligned} C_T \;-\; \tfrac{1}{2}\,\sigma^2(K,T)\,K^2\,C_{KK} \;+\; (r(T)-d(T))\,K\,C_K \;&=\; -\,d(T)\,C\,, && \\ & && (K,T)\in I\!R^+\times(0\,,\bar{T}] \\ C(\sigma(\cdot,\cdot);K,0) \;&=\; \max(S-K,0)\,, && K\in I\!R^+\,. \end{aligned}$$

This model can be thought as being dual to the first one since the variables in this model are given by the strike price K and the maturity T and the price of the underlying S and the current time t are hold fixed. Thus, we will call this second European call price model the dual extended Black–Scholes model. The coefficients in this second model are again nonconstant with one coefficient containing the volatility function $\sigma(K,T)$ depending on the strike price K and the maturity T which will be designated as the instantaneous volatility function.

The second model contains the volatility values that correspond to European call options with strike price K and maturity T, i.e. the volatility values that are necessary to evaluate European call options that are not traded in the market. Moreover, this second model has the advantage that a single evaluation of the model establishes the values of European call options for all conceivable strike prices and maturities for a fixed price of the underlying S and the current time t. Thus, in this thesis we will use the second model to determine the instantaneous volatility function $\sigma(K,T)$ necessary for a valuation of European call prices that is consistent with the market prices. This means that we will adapt the instantaneous volatility function in the dual extended Black–Scholes equation such that the model European call option values approximate their counterpart in the market. Thus, the mathematical formulation of the construction of the instantaneous volatility function represents an inverse problem with a system equation given by the dual extended Black–Scholes equation. These problems are naturally ill–posed.

The existence and uniqueness of a solution to the dual extended Black–Scholes equation is ensured by results given by Friedman [37] when the coefficients of the parabolic differential equation are sufficiently smooth (i.e. satisfy certain Hölder conditions). For the numerical solution of this initial–value problem, we are using a finite difference scheme. Since the numerical computations can only be carried out on a finite domain, we have to introduce boundary conditions such that the given initial–value problem is transformed into an initial–boundary value problem. The existence of a unique solution to this latter problem follows again from results given by Friedman [37].

The convergence of the solution of the finite difference scheme to the solution of the initial–boundary–value problem is derived by using the concepts of consistency and stability of the finite difference scheme. A main tool in proving the consistency of finite difference schemes is Taylor's expansion which requires a certain smoothness of the solution to the initial–boundary–value problem. However, for the given problem, this cannot be ensured since the initial condition in the initial–boundary–value problem is continuous but not differentiable everywhere. Ladyženskaja, Solonnikov, and Ural'ceva have shown in [66, IV, Theorem 5.2] that the smoothness of the solution to the parabolic initial–boundary–value problem depends on the smoothness of its initial condition. However, since the initial condition is evaluated at discrete points in the finite difference scheme, it is also possible that these discrete points present an approximation of a function possessing more smoothness features such as a certain order of differentiability. Hence, we will examine the consistency of the finite difference scheme with respect to a slightly perturbed initial–boundary–value problem that coincides with the original initial–boundary–value problem except for the initial condition which is exchanged with a sufficiently smooth approximation of the original initial condition. The maximum principle then ensures that the difference between the solution of these two initial–boundary–value problems is bounded by the maximal difference between the original initial condition and its approximation. The examination of the consistency of the slightly perturbed initial–boundary–value problem corresponds to the well–known ideas of consistency examinations for finite difference schemes applied to parabolic differential equations. The stability of the finite difference scheme is ensured under certain assumptions on the discrete coefficients of the parabolic differential equation and on the discretization sizes of the spatial and the time variable which also correspond to the known stability conditions for finite difference schemes for parabolic differential equations. Using these two results, we will show that the order of convergence of the solution of the finite difference scheme to the solution of the corresponding parabolic initial–boundary–value problem depends on the order of accuracy of a sufficiently smooth approximation of the nonsmooth initial condition of the original initial–boundary value problem. Moreover, we elucidate that similar problems occur also in the convergence analysis when finite element methods are applied to the solution of the initial–boundary–value problem.

Using a finite difference scheme for numerically solving the parabolic initial–boundary–value problem, the functional form of the volatility parameter in the inverse problem can be replaced by a finite dimensional approximation $\bar{\sigma}$ containing the values of the volatility function at the grid points of the finite difference mesh. Thus, the infinite dimensional inverse problem is replaced by a finite dimensional nonlinear least–squares problem. Because of the statements given above, this nonlinear least–squares problem is underdetermined such that the ill–posedness of the infinite dimensional inverse problem is carried over to its finite dimensional counterpart. For the solution of this problem, we again apply the inexact Gauss–Newton algorithm. A regularization of the ill–posedness of the problem is achieved because of the use of a trust–region strategy in the inexact Gauss–Newton algorithm. Using the Euclidean norm in the definition of the trust–region prevents the updates of the Gauss–Newton iterates from becoming too large. In the context of constructing the instantaneous volatility function, however, one wishes to determine the volatility function so that the dual extended Black–Scholes model approximates the values of the market European call options and which shows the most smooth behavior. Thus, in this case the trust–region is defined using a scaled norm with the scaling matrix given by a modified version of the discretized differentiation operator.

At the end of this chapter, we present some computational experience with the inexact Gauss–Newton method applied to the nonlinear least–squares problem describing the construction of the instantaneous volatility function. In the first example, we demonstrate the behavior of the algorithm by constructing the instantaneous volatility function for several synthetic European call options. In this example, we suppose that the instantaneous volatility function is known. Then, the market European call data are simulated by evaluating a set of European call options using a finite difference scheme applied to the parabolic differential equation containing the exact instantaneous volatility function. A comparison of the constructed with the exact instantaneous volatility function shows that the instantaneous volatility function is accurately approximated in regions where market European call option data are located. Finally, we examine a more realistic example by using values of market European DAX call options to approximate the instantaneous volatility function of the German stock index DAX.

Chapter 2

An Iterative Inexact Gauss–Newton Method

In this chapter, we are concerned with the solution of the large dimensional unconstrained nonlinear least–squares problem

$$\text{Minimize} \quad \phi(x) = \frac{1}{2}\, \|R(x)\|_2^2 \,, \quad x \in I\!R^n$$

where $R : I\!R^n \to I\!R^m$ and n is very large. In Section 2.1, we briefly introduce this class of problems and discuss their special structure. We are especially interested in underdetermined systems, where the number of variables is larger than the dimension of the range space, i.e. $n > m$. In this case, it is not surprising that $R(x)$ at the solution is fairly small. Therefore, we use a Gauss–Newton method for the solution of the nonlinear least–squares problem, which is described in Section 2.2. In Subsection 2.2.1, we introduce the basic ideas of the Gauss–Newton Method. In Chapter 3, we will show that the pure Gauss–Newton method converges only locally. To enlarge the region of convergence, we combine the Gauss–Newton method with a trust–region strategy. In Section 2.2.2, we describe the basic ideas of trust–region methods. In every step of the Gauss–Newton method combined with a trust–region strategy, a constrained linear least–squares problem

$$\text{Minimize} \quad \varphi(\delta x) = \frac{1}{2}\, \|J(x)\,\delta x + R(x)\|_2^2 \quad \text{subject to} \quad \|\delta x\|_{\bar{D}} \leq \Delta$$

has to be solved where $J(x) \in I\!R^{m \times n}$ is the Jacobian of the residual vector $R(x)$. The Jacobian $J(x)$ in the applications of Chapters 4 and 5 are not only large but also dense. Since subroutines for the evaluation of the matrix–vector products $J(x)\,\delta v$ and $J(x)^T \delta u$ are available for both applications, we use an iterative scheme for the solution of these constrained linear least–squares problems. In Subsection 2.2.3, we describe the modified [98] (truncated [18]) conjugate gradient method for the solution of the trust–region subproblems which was derived by Toint [105] and Steihaug [98]. At the end, we list the complete algorithm which we use for the solution of the nonlinear least–squares problems. This is a fully iterative algorithm which takes into account the dimension and the structure of the problems examined in Chapters 4 and 5 as well as their ill–posedness.

2.1 The Nonlinear Least–Squares Problem

Nonlinear least–squares problems are a special class of minimization problems where the objective function ϕ has the special form

$$\phi(x) = \tfrac{1}{2}\|R(x)\|_2^2 = \tfrac{1}{2}\sum_{i=1}^{m} R_i(x)^2 \tag{2.1.1}$$

with $R : I\!R^n \to I\!R^m$ is defined by

$$R(x) = (R_1(x)\,, R_2(x)\,, \ldots, R_m(x))^T\,. \tag{2.1.2}$$

The component functions R_i, $i = 1, \ldots, m$, are nonlinear real valued functions from $I\!R^n$ to $I\!R$, and we assume that $R_i(x)$, $i = 1, \ldots, m$, are sufficiently smooth. We refer to each of the R_i, $i = 1, \ldots, m$, as a residual and to R as the residual vector.

The nonlinear least–squares problem often arises in the context of data–fitting applications where one tries to fit a set of observation data with a model approximating the examined system. In these applications, usually, some a priori information is available to describe the examined system by a parameterized model. The unknown parameters of the model are then chosen such that the model matches as close as possible the output of the system at various observation points. Let out_i be the output of the system at the observation point y_i, $i = 1, \ldots, m$. Furthermore, suppose that the parameterized model of this system is described by the real function $z(x, y)$ which is nonlinear in the unknown parameters $x = (x_1, x_2, \ldots, x_n) \in I\!R^n$ and where y corresponds to the input variables of the system. If we define the ith residual R_i as the difference between the ith output out_i of the examined system and the output of the model evaluated at the ith observation point y_i, $i = 1, \ldots, m$,

$$R_i(x) = out_i - z(x, y_i)$$

then fitting the model to the data means to find a parameter vector x such that the residual vector R is minimized in an appropriate norm. Adapting the parameter vector x such that the sum of the squares of the residuals $R_i(x)$, $i = 1, \ldots, m$, is minimized, i.e. the Euclidean norm of R is minimized, is equivalent to solving a nonlinear least–squares problem. It is possible to use also a different norm for the measurement of the size of the residual vector R. However, the choice of the Euclidean norm is verified by statistical arguments when the errors are independently normally distributed. In this case, the minimization of the Euclidean norm of the residual vector R is equivalent to minimizing the variance [4], [95].

Generally, we could use any algorithm for unconstrained minimization problems for the solution of the nonlinear least–squares problem

$$\text{Minimize} \quad \phi(x)\,, \quad x \in I\!R^n \tag{2.1.3}$$

with $\phi(x)$ defined as in (2.1.1). However, as the gradient and the Hessian of $\phi(x)$ possess a certain structure these structural properties should be exploited and special algorithms adjusted to nonlinear least–squares problems should be used for their solution, see e.g. [21], [22], [85], [40, Section 4.7], [23], [25, Chapter 10], [24], [7, Chapter 9], [79, Chapter 10].

We define the Jacobian of the residual vector $R(x) = (R_1(x)\,, \ldots, R_m(x))^T \in I\!R^m$ by

$$J(x) = \left(\frac{\partial R_i(x)}{\partial x_j} \right)_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} \in I\!R^{m \times n}$$

and the Hessian of the ith residual function $R_i(x)\,, i = 1\,, \ldots, m$, by

$$H_i(x) = \nabla^2 R_i(x) = \left(\frac{\partial^2 R_i(x)}{\partial x_j\,\partial x_k} \right)_{j,k=1,\ldots,n} \in I\!R^{n \times n}\,, \quad i = 1\,, \ldots, m.$$

Then, the gradient and the Hessian of $\phi(x)$ can be written as:

$$\begin{aligned} \nabla \phi(x) &= J(x)^T R(x)\,, \\ \nabla^2 \phi(x) &= J(x)^T J(x) + \sum_{i=1}^{m} R_i(x) H_i(x)\,. \end{aligned} \tag{2.1.4}$$

Algorithms for the solution of nonlinear unconstrained minimization problems are based upon the approximation of the nonlinear objective function by a quadratic model function. In the case of the nonlinear least–squares problem (2.1.3), this quadratic model of the objective function $\phi(x)$ around a point x is defined by

$$\begin{aligned} m^q(x_+ - x) &= \phi(x) + \langle \nabla \phi(x), x_+ - x \rangle + \tfrac{1}{2}\, \langle x_+ - x, B\,(x_+ - x) \rangle \\ &= \tfrac{1}{2}\, \|R(x)\|_2^2 + \langle J(x)^T R(x), x_+ - x \rangle + \tfrac{1}{2}\, \langle x_+ - x, B\,(x_+ - x) \rangle \end{aligned} \tag{2.1.5}$$

where B is either the Hessian $\nabla^2 \phi(x)$ or some approximation of it. Note that the quadratic model will be a good approximation of the objective function only in a small neighborhood of the point x. The minimizer x_+ of the quadratic model $m^q(x_+ - x)$ serves as the new approximation of the solution x_* to the nonlinear least–squares problem (2.1.3). In case of a positive definite matrix B, the minimizer x_+ of (2.1.5) is the solution to the linear system of equations

$$B\,(x_+ - x) = -J(x)^T R(x)\,. \tag{2.1.6}$$

If Newton's method is used, then $B = \nabla^2 \phi(x)$ and the new approximation x_+ of the solution x_* to (2.1.3) is given by (if $\nabla^2 \phi(x)$ is positive definite)

$$\begin{aligned} x_+ &= x - B^{-1} J(x)^T R(x) \\ &= x - \left(J(x)^T J(x) + \sum_{i=1}^{m} R_i(x) H_i(x) \right)^{-1} J(x)^T R(x) \end{aligned}$$

This method exhibits a fast local q–quadratic rate of convergence if standard assumptions are fulfilled. However, to achieve this fast asymptotic convergence behavior it is necessary to compute the second derivatives $H_i(x)$ of the residual functions $R_i(x)$, $i = 1, \ldots, m$. Unfortunately, often they are either too expensive or even impossible to obtain. Nonlinear least–squares problems possess the special feature that the first part of the Hessian $\nabla^2 \phi(x)$ can be computed for free when the Jacobian $J(x)$ of the residual $R(x)$ is known. Moreover, the first part of $\nabla^2 \phi(x)$ is often more important than the second term which might be very small for points close to the solution x_*, especially when $R(x)$ is almost linear or very small at the solution x_*. This argument is considered in the Gauss–Newton method which we present in the next section.

2.2 The Inexact Gauss–Newton Method

2.2.1 The Gauss–Newton Method

So far, we have characterized the nonlinear least–squares problem as a special case of an unconstrained minimization problem. However, they are also conceptually close to the solution of a system of nonlinear equations (when $m = n$ the nonlinear least–squares problem includes the solution of a system of nonlinear equations as a special case) [25], [7]. In this case, a natural approach for the solution of the nonlinear least–squares problem would be the approximation of the residual vector $R(x)$ by the affine model

$$m^a(x_+ - x) = R(x) + J(x)\,(x_+ - x)\,. \tag{2.2.1}$$

and taking the solution x_+ of $m^a(x_+ - x) = 0$ as a new approximation of the solution to the nonlinear problem. In the special case of $m = n$, this is also the model usually used in algorithms for the solution of systems of nonlinear equations. Since in the general case $m \neq n$ we cannot guarantee the existence of a solution $x_+ - x$ to $m^a(x_+ - x) = 0$, a logical way is to find the point $x_+ - x$ which minimizes the Euclidean norm of the affine model, i.e. the solution to the linear least–squares problem

$$\text{Minimize} \quad \varphi(\delta x) = \tfrac{1}{2}\,\|R(x) + J(x)\,\delta x\|_2^2\,, \quad \delta x \in I\!R^n\,, \tag{2.2.2}$$

with $\delta x = x_+ - x$, see for instance Dennis and Schnabel [25, Chapter 10]. The resulting iterative process in which in every iteration the linear least–squares problem (2.2.2) is solved is called the Gauss–Newton method.

Since

$$\tfrac{1}{2}\,\|R(x) + J(x)\,\delta x\|_2^2 = \tfrac{1}{2}\,\langle R(x), R(x)\rangle + \langle J(x)^T R(x), \delta x\rangle + \tfrac{1}{2}\,\langle \delta x, J(x)^T J(x)\,\delta x\rangle$$

and $\|\cdot\|_2^2$ is a convex and differentiable function it follows that δx is a solution of the linear least–squares problem (2.2.2) if and only if δx is also a solution of the normal equations

$$J(x)^T J(x)\;\delta x = -J(x)^T R(x) \tag{2.2.3}$$

(see e.g. [18]).

In the previous section, we have shown that the Newton iterate is the solution of the following linear system of equations:

$$\left(J(x)^T J(x) + \sum_{i=1}^{m} R_i(x)\;H_i(x) \right)\;(x_+ - x) = -J(x)^T R(x)\,.$$

Since the linear least–squares problem (2.2.2) is equivalent to solving the normal equations (2.2.3), the difference between Newton's method and the Gauss–Newton method is that in the latter method, the second order terms in the Hessian of $\phi(x)$ are omitted. We will see in the next chapter that the success of the Gauss–Newton method will depend on the importance of the omitted term

$\sum_{i=1}^{m} R_i(x)\, H_i(x)$ in the Hessian $\nabla^2\phi(x)$. If, for instance, the residual functions $R_i(x)$ are small near the solution x_* then the second order term in the Hessian of $\phi(x)$ is small and the approximation $J(x)^T J(x)$ of the Hessian of $\phi(x)$ in the Gauss–Newton method is close to the Hessian $\nabla^2\phi(x)$ itself [15], [25]. If the residual vector $R(x_*)$ at the solution x_* is zero, then the Gauss–Newton method even achieves similar good asymptotic convergence results than Newton's method.

The applications in Chapters 4 and 5 are characterized by the fact that the number of the residual functions R_i, $i = 1, \dots, m$, is smaller than the number of variables x_j, $j = 1, \dots, n$, i.e. the systems are underdetermined. Thus, it is not surprising that the residual functions in these cases tend to be fairly small at the solution x_* such that the Gauss–Newton method seems to be a reliable method for the solution of these nonlinear least–squares problems [15].

Since in the general case $m \neq n$ the solution to the normal equations (2.2.3) is non–unique, it is to be expected that this leads to numerical instabilities. To account for the local validity of the affine model $m^a(\delta x)$ and the characteristics of the problems in Chapters 4 and 5 the solution of the normal equations is taken with minimal norm. This solution is given by

$$\delta x = x_+ - x = -J(x)^+ R(x) \tag{2.2.4}$$

where $J(x)^+$ is the Moore–Penrose inverse (or pseudoinverse) of the Jacobian $J(x)$ of the residual vector $R(x)$. In the following, we will refer to this direction as the Gauss–Newton direction. If $m > n$ and the Jacobian $J(x)$ has full column rank, then the Moore–Penrose inverse is given by

$$J(x)^+ = (J(x)^T J(x))^{-1} J(x)^T$$

and hence, in this special case the Gauss–Newton direction is given by (see e.g. [25, Section 10.2])

$$x_+ - x = -(J(x)^T J(x))^{-1} J(x)^T R(x)\,.$$

In the following Lemma 2.2.1, we show that the Gauss–Newton direction is a descent direction, i.e. $-\nabla\phi(x)^T J(x)^+ R(x) < 0$, as long as x is not a critical point. Note that a point x_* is called a *critical point* if x_* satisfies the first–order necessary optimality condition for a minimizer, i.e. in case of the nonlinear least–squares problem (2.1.3) a critical point x_* satisfies $\nabla\phi(x_*) = J(x_*)^T R(x_*) = 0\,$.

Lemma 2.2.1 *Let x be not a critical point, i.e. $J(x)^T R(x) \neq 0$. Then, the Gauss–Newton direction $\delta x = -J(x)^+ R(x)$ is a descent direction, i.e. $\nabla\phi(x)^T \delta x < 0\,$.*

Proof: For the proof of Lemma 2.2.1, we have to verify that (see (2.1.4) for the special form of $\nabla\phi(x)$)

$$(J(x)^T R(x))^T (-J(x)^+ R(x)) = -R(x)^T J(x)\, J(x)^+ R(x) < 0$$

Since $J(x)^+$ is the Moore–Penrose inverse of $J(x)$ the following identities hold:

$$J(x)^+ = J(x)^+ J(x)\; J(x)^+ \quad \text{and} \quad J(x)\; J(x)^+ = (J(x)\; J(x)^+)^T\,.$$

Hence,

$$\begin{aligned}
-R(x)^T J(x)\, J(x)^+ R(x) &= -R(x)^T J(x)\, J(x)^+ J(x)\, J(x)^+ R(x) \\
&= -R(x)^T (J(x)\, J(x)^+)^T J(x)\, J(x)^+ R(x) \\
&= -(J(x)\, J(x)^+ R(x))^T J(x)\, J(x)^+ R(x) \\
&= -\|J(x)\, J(x)^+ R(x)\|_2^2 \\
&\leq 0
\end{aligned}$$

The identity of the null spaces of $J(x)^T$ and $J(x)^+$, i.e. $\mathcal{N}(J(x)^T) = \mathcal{N}(J(x)^+)$), yields $-J(x)^+ R(x) \neq 0$. Furthermore, the Gauss–Newton direction $\delta x = -J(x)^+ R(x)$ is orthogonal to the null space $\mathcal{N}(J(x))$ of the Jacobian $J(x)$, i.e. $J(x)^+ R(x) \perp \mathcal{N}(J(x))$ such that $J(x)\; J(x)^+ R(x) \neq 0$. Thus,

$$\nabla\phi(x)^T \delta x = -\|J(x)\; J(x)^+ R(x)\|_2^2 < 0\,.$$

■

2.2.2 Globalization of the Algorithm: Introduction of a Trust–Region Strategy

We will show in Section 3.2 that the Gauss–Newton method converges only locally. In addition, it is possible that for ill–posed problems the Gauss–Newton direction is almost orthogonal to the negative gradient such that it might be a bad descent direction. Furthermore, the performance of the Gauss–Newton method is highly sensitive to the estimate of the rank of the Jacobian, see e.g. Gill, Murray and Wright [40, Example 4.11]. This results from the fact that the Moore–Penrose inverse varies discontinuously when the rank of the matrix changes, see Wedin [106], Björck [7, Section 1.4.2]. One way to deal with these difficulties is the introduction of a trust–region strategy which yields a globalization of the overall algorithm as well as a regularization of the iteration. A globalization of the algorithm might also be achieved by using a line search method. However, in case of a rank deficient Jacobian, line search methods always must include some estimation strategy for the rank of the Jacobian to compute a reliable Gauss–Newton direction [7, Section 9.2.1]. This is not necessary when using a trust–region strategy since this method enforces a regularization of the subproblem when the Gauss–Newton direction has bad descent providing a self adapting procedure to handle the situation mentioned above.

The trust–region strategy consists in the introduction of a bound Δ on the size of the variable vector in the quadratic Gauss–Newton subproblem. This bound also guarantees that the quadratic approximation of the nonlinear least–squares problem will be restricted to a certain neighborhood of the current iterate. Hence, we have to solve the problem

$$\text{Minimize} \quad \varphi_i(\delta x) = \tfrac{1}{2}\, \|R(x_i) + J(x_i)\, \delta x\|_2^2 \quad \text{subject to} \quad \|\delta x\|_{\bar{D}} \leq \Delta \tag{2.2.5}$$

in each step of the algorithm, where $\|\delta x\|_{\bar{D}} = \sqrt{\langle x, \bar{D}\, x\rangle}$ with $\bar{D} \in I\!R^{n\times n}$ is a symmetric positive definite matrix. In the case when $\bar{D} = I$, with $I \in I\!R^{n\times n}$ is the n–dimensional identity matrix, this is just the ℓ_2 norm. However, if the iterates δx_k are badly scaled numerical instabilities can occur. In this case, the problem might be stabilized by introducing a scaling matrix $\bar{D}$ (see e.g. Conn, Gould and Toint [18, Section 6.7]). It is also possible that the choice of the norm in (2.2.5) results from the characteristics of the specific problem (see for instance the application in Chapter 5). If the resulting trust–region subproblems are formulated in the original variables, then they have the form as in (2.2.5).

In the following, we examine the general case where $\bar{D} \in I\!R^{n\times n}$ is an arbitrary symmetric positive definite matrix. Then, the basic trust–region strategy applied to the Gauss–Newton method has the following structure (see for instance Conn, Gould, and Toint [18, Section 6.1]):

Algorithm 2.2.1 (Trust–Region Method)

```
Let an initial approximation $x_0$, and an initial trust-region
radius $\Delta_0 \le \bar{\Delta}$ be given and set $0 < a_1 \le a_2 < 1$, and $0 < \gamma_3 \le \gamma_4 < 1 \le \gamma_5$.
Compute $\phi(x_0)$ and set $i = 0$.
1.) Find an approximate solution $\delta x_i$ to

        Minimize  $\varphi_i(\delta x) = \frac{1}{2}\ \|R(x_i) + J(x_i)\,\delta x\|_2^2$  s.t.  $\|\delta x\|_{\bar{D}} \le \Delta_i$

2.) Compute $\phi(x_i + \delta x_i)$ and

        $\rho_i = \dfrac{\phi(x_i) - \phi(x_i + \delta x_i)}{\varphi_i(0) - \varphi_i(\delta x_i)}$.

3.) Set

        $x_{i+1} = \begin{cases} x_i + \delta x_i & , \text{ if } \rho_i \ge a_1, \\ x_i & , \text{ otherwise.} \end{cases}$

4.) Determine

        $\Delta_{i+1} \in \begin{cases} [\Delta_i\,,\ \min\{\gamma_5\,\Delta_i,\ \bar{\Delta}\}] & , \text{ if } \rho_i \ge a_2\,, \\ [\gamma_4\,\Delta_i\,,\ \Delta_i] & , \ \rho_i \in [a_1\,, a_2)\,, \\ [\gamma_3\,\Delta_i\,,\ \gamma_4\,\Delta_i] & , \ \rho_i < a_1 \end{cases}$

5.) If $\phi(x_{i+1}) \le \epsilon$ or $\|\nabla\phi(x_{i+1})\| = \|J(x_{i+1})^T R(x_{i+1})\| \le \epsilon$ stop,
    else set $i := i + 1$ and go to step 1.
```

Note that $\phi(x) = \frac{1}{2}\ \|R(x)\|_2^2$ is the objective function of the original nonlinear least–squares problem. In Step 1 of this algorithm, we compute an approximate minimizer to our quadratic model problem. The minimization is restricted to the neighborhood of the current iterate x_i where we can *trust* the quadratic model function, i.e. where the quadratic model function approximates the original objective function sufficiently good. We call the domain

$$\mathcal{U}_i = \{\ x \in I\!R^n \mid x = x_i + \delta x\,,\ \|\delta x\|_{\bar{D}} \leq \Delta_i\ \} \tag{2.2.6}$$

the trust region and the value Δ_i the trust–region radius. In the second step the quotient ρ_i is computed which gives information about the ratio of the decrease in the objective function $\phi(x)$ to the decrease in the model $\varphi_i(\delta x)$. If this quotient is sufficiently close to 1, i.e. the reduction predicted by the model coincides with the reduction in the objective function then the trial step computed in Step 1 is accepted (see Step 3). On the other side, if the reduction in the model does not estimate the reduction in the objective function well enough, the trial step from Step 1 of Algorithm 2.2.1 is rejected and the trust–region radius Δ_i is reduced in Step 4 of the routine in hopes that the quadratic model can be trusted more in the smaller region.

Although we only need to compute an approximate solution to problem (2.2.5) in Step 1 of Algorithm 2.2.1, we firstly examine the global solution of (2.2.5) which is characterized in the following Theorem 2.2.2 (see for instance Moré [76, Theorem 3.11], Conn, Gould and Toint [18, Section 7.4])

Theorem 2.2.2 *The vector δx_i is a global minimizer to the trust–region subproblem* (2.2.5) *if and only if $\|\delta x_i\|_{\bar{D}} \leq \Delta$ and*

$$(J(x_i)^T J(x_i) + \lambda_i\, \bar{D})\delta x_i = -J(x_i)^T R(x_i) \tag{2.2.7}$$

for some $\lambda_i \geq 0$ and $\lambda_i(\|\delta x_i\|_{\bar{D}} - \Delta) = 0$. Moreover, if $J(x_i)^T J(x_i) + \lambda_i\, \bar{D}$ is positive definite then δx_i is unique.

This theorem was independently proved by Gay [38] and Sorensen [97] and it is based on a result by Goldfeld, Quandt and Trotter [41]. The representation (2.2.7) of the global solution is a direct consequence from the first–order necessary optimality conditions for constrained optimization (also known as the Karush–Kuhn–Tucker (KKT) conditions).

Theorem 2.2.2 also makes clear the close relationship between the Gauss–Newton method combined with a trust–region strategy and the well–known Levenberg–Marquardt method [69], [72]. The difficulty in the Levenberg–Marquardt method is the choice of the parameter λ_i. Moré presented in [75] an implementation of the Levenberg–Marquardt method (2.2.7) in a trust–region framework. The advantage of the trust–region method in comparison with the Levenberg–Marquardt method is the explicit control of the size of the steps.

Moreover, from Theorem 2.2.2 follows that either the minimizer of the unconstrained linear least–squares problem (2.2.2) lies in the inside of the trust–region or the trust–region boundary is active and the minimizer of the constrained linear least–squares problem (2.2.5) occurs on the boundary.

Finally, (2.2.7) elucidates the regularization property of the trust–region strategy.

2.2.3 Iterative Solution of the Trust–Region Subproblem

The central computation in Algorithm 2.2.1 is the calculation of the approximate solution to the constrained linear least–squares problems in Step 1. To obtain global convergence of the Trust–Region Method 2.2.1, it is necessary to guarantee a sufficient reduction in the model $\varphi_i(\delta x)$, for $x_i + \delta x \in \mathcal{U}_i$, see Conn, Gould, and Toint [18, Section 6.3]. We will see that the Cauchy point already satisfies such a sufficient decrease condition.

Definition 2.2.3 Let δx_i^{sd} be the steepest descent direction to the trust–region subproblem (2.2.5), and let ϑ_i^{cp} be the optimal solution to

$$\min_{\vartheta > 0} \; s(\vartheta) = \varphi(-\vartheta \, \delta x_i^{sd}) \quad \text{subject to} \quad \|\vartheta \, \delta x_i^{sd}\|_{\bar{D}} \leq \Delta_i$$

then

$$x_i^{cp} = x_i - \vartheta_i^{cp} \, \delta x_i^{sd} = x_i + \delta x_i^{cp}$$

is called the Cauchy point and δx_i^{cp} the Cauchy direction.

The Cauchy direction has the advantage that it is cheap to compute; however, the repeated use of this direction might result in a poor convergence behavior of the method. On the other side, the global minimizer of the trust–region subproblem, which is characterized in Theorem 2.2.2, might lead to an asymptotically fast rate of convergence but the computational effort of this point might be very high (see Conn, Gould, and Toint [18, Sections 7.3 – 7.4] for finding the ℓ_2–norm and the scaled ℓ_2–norm model minimizer).

Since the dimension of the problems in Chapters 4 and 5 is very large, we are especially interested in efficient solution methods for the trust–region subproblems. Additionally, we cannot fall back upon a particular structure of the Jacobian appearing in the problems in Chapters 4 and 5 such that we have to suppose that $J(x)$ is not only large but also dense. Thus, it is impossible to store the matrix $J(x)$ or even $J(x)^T J(x)$. However, for both applications we can make efficient subroutines for the evaluation of the matrix–vector products $J(x)\delta v$ and $J(x)^T \delta u$ available (see Lemma 4.2.1, Lemma 4.2.2 and Theorem 5.5.2, Theorem 5.5.3, respectively). A promising method for the solution of the unconstrained version of the linear least–squares problem (2.2.2) that must not know $J(x)$ explicitly and which makes only use of the mentioned matrix–vector products is the conjugate gradient method due to Hestenes and Stiefel [56] respective its preconditioned version. A further benefit of the conjugate gradient method is that it converges to the minimal norm solution of the unconstrained linear least–squares problem (2.2.2) if the starting element is chosen properly (see e.g. Hestenes [54, Section 4], [55, p. 296]).

Toint [105] and Steihaug [98] modified the preconditioned conjugate gradient method so that it is applicable to quadratic trust–region subproblems with a symmetric approximation of the Hessian. This algorithm finds an approximate solution to the trust–region subproblem and generates a compromise between the Cauchy direction and the global minimizer of the subproblem.

Applying the conjugate gradient method to problem (2.2.5) with the trust–region being defined using the Euclidean norm $\| \cdot \|_2$, i.e. $\bar{D} = I$, then the length of the iterates δx_k, i.e. $\|\delta x_k\|_2$, is strictly increasing with k, see Lemma 2.2.6. This characteristic ensures that the iterates will not return in the trust–region if they have once left this area. Thus, the conjugate gradient method can be stopped when the conjugate gradient iterates become too large (this will be one of the modifications introduced in the algorithm presented by Toint [105] and Steihaug [98]). However, if the length of the iterates is measured in the $\bar{D}$–norm $\| \cdot \|_{\bar{D}}$, $\bar{D} \neq I$, this monotonicity characteristic is not guaranteed when the conjugate gradient method is applied to problem (2.2.5). If we set $z = \bar{D}^{1/2} \delta x$, then, problem (2.2.5) can be easily transformed to

$$\text{Minimize} \quad \varphi_i(z) = \tfrac{1}{2} \, \|R(x_i) + J(x_i) \, \bar{D}^{-1/2} z\|_2^2 \quad \text{subject to} \quad \|z\|_2 \leq \Delta \tag{2.2.8}$$

such that the $\bar{D}$–norm problem (2.2.5) is equivalent to a trust–region subproblem with the trust–region being defined using the Euclidean norm. Now, applying the conjugate gradient method to problem (2.2.8) and transforming the variables back to the variables of the original problem (2.2.5), then yields the preconditioned version of the conjugate gradient method with $\bar{D}^{1/2}$ being the preconditioner.

This modified [98] (truncated [18]) preconditioned conjugate gradient method presented by Steihaug respective Toint has the following form:

Algorithm 2.2.2 (Modified Preconditioned Conjugate Gradient Method)

```
Set initially $\delta x_0 = 0$, $r_0 = -R(x)$, solve $\bar{D}^{1/2} p_0 = -J(x)^T R(x)$,
and set $s_0 = p_0$.  Given $\xi > 0$ and $\Delta > 0$.
For $k = 0, 1, 2, \ldots, n$ do
  1.) Solve   $\bar{D}^{1/2} v_k = p_k$
  2.) Set     $\psi_k = \|J(x) \, v_k\|_2^2$
  3.) If      $\psi_k = 0$ set $\delta x_{k+1} = \delta x_k$, and stop
      Else    continue with Step 4.
  4.) Set     $\alpha_k = \|s_k\|_2^2 / \psi_k$
  5.) Set     $\delta x_{k+1} = \delta x_k + \alpha_k \, v_k$
  6.) If      $\|\delta x_k + \alpha_k v_k\|_{\bar{D}} \geq \Delta$
              compute $\tau_k > 0$ so that $\|\delta x_k + \tau_k v_k\|_{\bar{D}} = \Delta$,
              set $\delta x_{k+1} = x_k + \tau_k v_k$, and stop
      Else    continue with Step 7.
  7.) Set     $r_{k+1} = r_k - \alpha_k \, J(x) \, v_k$
  8.) Solve   $\bar{D}^{1/2} s_{k+1} = J(x)^T r_{k+1}$
```

Algorithm 2.2.2 (Modified Precond. Conjugate Gradient Method, cont.)

```
 9.)  If   $\|s_{k+1}\|_2 \leq \xi\ \|J(x)^T R(x)\|_{\bar{D}}$,   stop
10.)  Set  $\beta_k = \|s_{k+1}\|_2^2 / \|s_k\|_2^2$
11.)  Set  $p_{k+1} = s_{k+1} + \beta_k\, p_k$
```

Note that in this algorithm, the first iterate δx_1 is just the Cauchy direction since we set initially $\delta x_0 = 0$.

The single steps of Algorithm 2.2.2 are those of the preconditioned version of the conjugate gradient method applied to linear least–squares problems with $\bar{D}^{1/2}$ chosen as the preconditioner, see for instance Björck [7, Algorithm PCCGLS]. Since we are examining the general case of trust–region subproblems with scaled $\bar{D}$–norm trust–regions, we use the preconditioned version of the conjugate gradient method, see Conn, Gould, and Toint [18, Section 6.7 and 7.4] and the statements about the equivalence of the trust–region subproblems (2.2.5) and (2.2.8) for the connection of trust–region scaling and preconditioning of the conjugate gradient iterations.

The modification in Algorithm 2.2.2 consists in the introduction of two further stopping criteria to account for the applicability of the algorithm also to positive semidefinite systems and the observance of the trust–region restriction. If in Step 3 of Algorithm 2.2.2 the direction v_k lies in the null space of $J(x)$, i.e. $\psi_k = 0$, then the value of the model $\varphi(\delta x_k + \alpha\, v_k)$ stays constant along this line since

$$\varphi(\delta x_k + \alpha\, v_k) \;=\; \tfrac{1}{2}\, \|R(x) + J(x)\,(\delta x_k + \alpha\, v_k)\|_2^2 \;=\; \tfrac{1}{2}\, \|R(x) + J(x)\,\delta x_k\|_2^2 \;=\; \varphi(\delta x_k)\,.$$

In this case, we set $\delta x_{k+1} = \delta x_k$ and stop the algorithm. It is also possible to compute the positive root τ_k to $\|\delta x_k + \tau_k\, v_k\|_{\bar{D}} = \Delta$ to obtain an approximate solution on the boundary of the trust–region; however, since the value of φ does not change along this line and we are not interested in steps which are too large (see the arguments in Chapters 4 and 5) we terminate the algorithm with the current iterate.

The following Lemma 2.2.4 shows that this stopping criteria is not necessary if $\bar{D}^{1/2} = I$, with I is the identity matrix. In this case, it is $v_k = p_k$ and $J(x)\, p_k \neq 0$ which is a simple consequence of the formula for p_k and the orthogonality of the range space of $J(x)^T$ to the null space of $J(x)$.

Lemma 2.2.4 *Let $\bar{D}^{1/2} = I$ and p_k, $k = 0, \ldots, j$, be generated in Step 11 of Algorithm 2.2.2. Then,*

$$J(x)\, p_k \neq 0 \quad \textit{for} \quad k = 0, \ldots, j\,.$$

Proof: From Step 11 of Algorithm 2.2.2 together with the initialization $p_0 = -J(x)^T R(x)$, it follows with $\bar{D}^{1/2} = I$

$$\begin{aligned} p_k &= s_k + \beta_{k-1}\, p_{k-1} \\ &= J(x)^T\, r_k + \beta_{k-1}\, (s_{k-1} + \beta_{k-2}\, p_{k-2}) \\ &= J(x)^T\, r_k + \sum_{i=0}^{k-1} \left(\prod_{j=i}^{k-1} \beta_j \right) J(x)^T r_i \\ &= J(x)^T \left(r_k + \sum_{i=0}^{k-1} \left(\prod_{j=i}^{k-1} \beta_j \right) r_i \right) \end{aligned}$$

and, thus, $p_k \in \mathcal{R}(J(x)^T)$. Furthermore, it follows that

$$\begin{aligned} \|p_k\|_2^2 &= \|s_k + \beta_{k-1}\, p_{k-1}\|_2^2 \\ &= \|s_k\|_2^2 + \beta_{k-1}^2\, \|p_{k-1}\|_2^2 + 2\, \beta_{k-1} \langle s_k\, , p_{k-1} \rangle \\ &= \|s_k\|_2^2 + \beta_{k-1}^2\, \|p_{k-1}\|_2^2 \end{aligned}$$

On this occasion, we have used the orthogonality of the gradient s_k and the conjugate directions p_{k-1} in the conjugate gradient method, see for instance Conn, Gould, and Toint [18, Lemma 5.1.4]. Note that $\|s_k\|_2 \neq 0$ since otherwise the Modified Preconditioned Conjugate Gradient Method 2.2.2 would have been stopped in a previous iteration (see Step 9 of Algorithm 2.2.2). Thus, $p_k \neq 0$. Since $\mathcal{R}(J(x)^T) = \mathcal{N}(J(x))^{\perp}$ it follows that $J(x)\, p_k \neq 0$. ■

Remark 2.2.5 Since δx_k in step 5 of Algorithm 2.2.2 can be written as

$$\delta x_k = \delta x_{k-1} + \alpha_{k-1}\, v_{k-1} = \delta x_0 + \sum_{i=0}^{k-1} \alpha_i\, v_i$$

from Lemma 2.2.4 follows that $\delta x_k \in \mathcal{R}(J(x)^T)$ if the initial point δx_0 lies in this space and if $\bar{D}^{1/2} = I$ (see also [18, Theorem 5.3.1]).

The second stopping criterion occurs in Step 6 of the Algorithm 2.2.2. Here, we test if the new iterate of the preconditioned conjugate gradient method stays within the trust–region. If the length of the new iterate is larger than the trust–region radius, we compute a linear combination of the current and the new iterate of the preconditioned conjugate gradient method which just reaches to the boundary of the trust–region and stop the algorithm. The following lemma, which is proved by Steihaug [98], ensures that subsequent iterates will not return in the interior of the trust–region if a former iterate has left this region.

Lemma 2.2.6 *Let δx_k, $k = 0, \dots, j$, be the iterates generated by the Modified Preconditioned Conjugate Gradient Algorithm 2.2.2, and let $\delta \hat{x}$ be the terminating iterate of the algorithm. Then the iterates δx_k, $k = 0, \dots, j$, satisfy the following two conditions:*

(a) The model function $\varphi(\delta x_k)$ is strictly decreasing for $k = 0, \ldots, j$, and

$$\varphi(\delta\hat{x}) \;\le\; \varphi(\delta x_j)\,. \tag{2.2.9}$$

(b) The length of the iterates $\|\delta x_k\|_{\bar{D}}$ is strictly increasing for $k = 0, \ldots, j$, and

$$\|\delta\hat{x}\|_{\bar{D}} \;\ge\; \|\delta x_j\|_{\bar{D}}\,. \tag{2.2.10}$$

Thus, if Algorithm 2.2.2 has generated an iterate that lies on the boundary of the trust–region it follows from Lemma 2.2.6 that the global minimizer of the model $\varphi_i(\delta x)$ is not in the interior of the trust–region.

Moreover, since the first iterate generated by the Modified Preconditioned Conjugate Gradient Algorithm 2.2.2 is the Cauchy point, which already satisfies a necessary sufficient decrease condition (see for instance Powell [84, Theorem 4]), and since every subsequent iterate of the algorithm achieves a larger reduction of the model function $\varphi_i(\delta x)$ according to Lemma 2.2.6(a), it follows that every iterate generated by Algorithm 2.2.2 decreases the model function sufficiently. This aspect will ensure the convergence of Algorithm 2.2.1 to a first–order critical point if Algorithm 2.2.2 is used for the solution of the trust–region subproblems (2.2.5) in Step 1 of Algorithm 2.2.1.

Next, we analyze the residual $\hat{r}$ of the linear model $J(x)\,\delta x + R(x)$ in the linear least–squares problem which corresponds to the terminating point $\delta\hat{x}$ of the Algorithm 2.2.2. It is well known that for r_{k+1} in Step 7 of Algorithm 2.2.2 it holds that ($\delta x_{k+1} = \sum_{j=0}^{k} \alpha_j v_j$ if $\delta x_0 = 0$)

$$r_{k+1} = r_k - \alpha_k J(x) v_k = r_0 - J(x) \sum_{j=0}^{k} \alpha_j v_j = -(R(x) + J(x)\,\delta x_{k+1})\,.$$

If the modified preconditioned conjugate gradient method 2.2.2 is stopped because $\psi_k = 0$ (Step 3) or the k-th conjugate gradient iterate δx_k violates the trust region bound (Step 6), then the same relationship holds with $\alpha_k = \tau_k$. Thus,

$$\hat{r} = -(J(x)\,\delta\hat{x} + R(x)). \tag{2.2.11}$$

If the Modified Preconditioned Conjugate Gradient Method 2.2.2 terminates neither in Step 3 nor in Step 6, then the following theorem gives information about the rate of convergence of the residuals $r_k = J(x)\;\delta x_k + R(x)$ which are computed in Step 7 of Algorithm 2.2.2.

Theorem 2.2.7 *Let $\{\delta x_k\}$ be generated by Algorithm 2.2.2. Then, for the residuals $r_k = J(x)\,\delta x_k + R(x)$ computed in Step 7 of Algorithm 2.2.2, we have*

$$\|r_k - r_*\|_2 \;\le\; \min_{p\in\mathcal{P}_k} \; \max_{\sigma\in\{\sigma_1,\ldots,\sigma_q\}} \; p(\sigma^2)\;\|r_0 - r_*\|_2 \tag{2.2.12}$$

where $r_* = J(x)\,\delta x_* + R(x)$, $\delta x_* = -J(x)^+ R(x)$, $\mathcal{P}_k$ is the set of polynomials p of degree k or less with $p(0) = 1$ and $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_q$ are the singular values of $J(x)\,\bar{D}^{-1/2}$.

(See e.g. Nachtigal, Reddy and Trefethen [77, Theorem 1] or Greenbaum [45].)

An error bound for the right side of the inequality in (2.2.12) is given by

$$\frac{\|r_k - r_*\|_2}{\|r_0 - r_*\|_2} \;\leq\; \min_{p \in \mathcal{P}_k} \; \max_{\sigma \in \{\sigma_1, \ldots, \sigma_q\}} \; p(\sigma^2) \;\leq\; 2 \left(\frac{\kappa(J(x)\,\bar{D}^{-1/2}) - 1}{\kappa(J(x)\,\bar{D}^{-1/2}) + 1} \right)^k \tag{2.2.13}$$

where $\kappa(J(x)\,\bar{D}^{-1/2}) = \sigma_1/\sigma_q$ is the condition number of $J(x)\,\bar{D}^{-1/2}$ (see for instance Greenbaum [45, Theorem 3.1.1] or Conn, Gould and Toint [18, Theorem 5.1.8]).

The inequality in (2.2.13) shows that the convergence of the preconditioned conjugate gradient method depends on the condition number of the matrix $J(x)\,\bar{D}^{-1/2}$, i.e. the convergence of the conjugate gradient method is slower if the condition number of $J(x)\,\bar{D}^{-1/2}$ is large. Furthermore, by choosing an appropriate polynomial $p(\sigma)$ in (2.2.12) (see e.g. Kelley [64, Section 2.2] or Björck [7, Section 7.4.2]) it follows that the reduction in the residuals will be larger if the singular values of $J(x)\,\bar{D}^{-1/2}$ are clustered. Thus, an appropriate choice of a preconditioner $\bar{D}^{-1/2}$ can yield an acceleration of the convergence of the preconditioned conjugate gradient method.

However, note that the acceleration of the convergence is combined with an increase of the computational cost. If $\bar{D}^{-1/2} \neq I$, then in each iteration of Algorithm 2.2.2 the linear systems of equations in Steps 1 and 8 have to be solved. The cost for the solution of these linear systems depends on the form of the preconditioner $\bar{D}^{-1/2}$. If $\bar{D}^{-1/2}$ is a diagonal matrix then the additional effort consists in $2n$ multiplications. In case $\bar{D}^{-1/2}$ is a lower or an upper triangular matrix then the linear system can be computed by forward or back substitution [43], respectively. However, if $\bar{D}^{-1/2}$ has a more general structure, then the additional cost consists in factorizing the matrix $\bar{D}^{-1/2}$ and in solving the systems in Step 1 and 8 using this factorization. However, a wise choice of the preconditioner may lead to a large reduction of the number of iterations in Algorithm 2.2.2 such that the overall effort of the preconditioned version might be smaller than the effort for the version without the preconditioner.

Unfortunately in general, it can be difficult to find an efficient preconditioner. In the problems arising from the applications examined in Chapters 4 and 5, we have the difficulty that we do not know the Jacobian $J(x)$ explicitly and that we expect a dense structure of this matrix. Thus, we do not have sufficient information to built an efficient preconditioner for these problems.

However, the problem formulation in Chapter 5 leads to a scaling of the iterates δx_k with a modified discrete version of the differentiation operator. Although this scaling will not be a preconditioning in the original sense we will use the preconditioned version of the conjugate gradient method for the solution of the resulting linear least–squares problems.

Thus, the complete algorithm we will use for the solution of the nonlinear least–squares problems appearing in Chapters 4 and 5 has the following form:

Algorithm 2.2.3 (Inexact Gauss–Newton Method)

```
Let
    an initial approximation $x_0$,
    an initial trust-region radius $\Delta_0 \le \bar{\Delta}$, and
    an initial relative error $\xi_0$ for the algorithm in
    Step 1
be given and set $0 < a_1 \le a_2 < 1$, and $0 < \gamma_3 \le \gamma_4 < 1 \le \gamma_5$.
Compute $\phi(x_0)$ and $R(x_0)$ and set $i = 0$.
1.) Use the Modified Preconditioned Conjugate Gradient
    Algorithm 2.2.2 to find an approximate solution $\delta x_i$
    to
        Minimize   $\varphi_i(\delta x) = \frac{1}{2} \, \|R(x_i) + J(x_i)\, \delta x\|_2^2$
        subject to $\|\delta x\|_{\bar{D}} \le \Delta_i$
    with relative error $\xi_i$.
2.) Compute $\phi(x_i + \delta x_i)$
        $\rho_i = \frac{\phi(x_i) - \phi(x_i + \delta x_i)}{\varphi_i(0) - \varphi_i(\delta x_i)}$.
3.) Set
        $x_{i+1} = \begin{cases} x_i + \delta x_i & , \text{ if } \rho_i \ge a_1, \\ x_i & , \text{ otherwise.} \end{cases}$
4.) Determine
        $\Delta_{i+1} \in \begin{cases} [\Delta_i\,, \min\{\gamma_5\,\Delta_i, \bar{\Delta}\}] & , \text{ if } \rho_i \ge a_2\,, \\ [\gamma_4\,\Delta_i\,, \Delta_i] & , \rho_i \in [a_1\,, a_2)\,, \\ [\gamma_3\,\Delta_i\,, \gamma_4\,\Delta_i] & , \rho_i < a_1 \end{cases}$
5.) Compute $\phi(x_{i+1})$ and $R(x_{i+1})$ and update $\xi_{i+1}$.
    If $\phi(x_{i+1}) \le \epsilon$ or $\|\nabla\phi(x_{i+1})\| = \|J(x_{i+1})^T R(x_{i+1})\| \le \epsilon$ stop,
    else set $i := i + 1$ and go to step 1.
```

The relative error ξ_i controls the accuracy of the approximate solution of the linear least–squares subproblem. Since additional stopping criteria are integrated in the preconditioned conjugate gradient method, the direction generated deviates from the original Gauss–Newton direction. Thus, we call the algorithm an inexact Gauss–Newton method.

At the beginning of the iteration process, it is not necessary to compute the approximate solution in Step 1 of Algorithm 2.2.3 to a high accuracy. However, when the Gauss–Newton iterates approach a critical point an increased accuracy is necessary to improve the asymptotical convergence behavior (see Section 3.2).

Chapter 3

Convergence

In this chapter, we examine the global convergence as well as the local convergence rate of the algorithm presented in Chapter 2. The global convergence of the inexact Gauss–Newton method results from the introduction of the trust–region strategy into the algorithm. Since the global convergence theory of trust–region methods is already well established we outline briefly the main assumptions in Section 3.1, which are necessary to ensure the global convergence. There exists also a variety of results about the local convergence behavior of the Gauss–Newton method. However, in the analysis of the asymptotic rate of convergence, usually, it is assumed that the residual function $R(x)$ is overdetermined and that the Jacobian $J(x)$ has full column rank. But, the applications we treat in Chapters 4 and 5 are characterized by the fact that the dimension of the variable is much larger than the dimension of the range space such that in these cases $R(x)$ is underdetermined. Therefore, in Section 3.2, we examine the local rate of convergence of the inexact Gauss–Newton method for underdetermined as well as overdetermined systems.

3.1 Global Convergence of the Inexact Gauss–Newton Method

In the following, we examine if the sequence of iterates $\{x_i\}$ generated by Algorithm 2.2.3, starting with an arbitrary initial element x_0, will converge to a limit point x_* that is a critical point to the nonlinear least–squares problem, i. e. that satisfies

$$\nabla\phi(x_*) \;=\; J(x_*)^T R(x_*) \;=\; 0\,.$$

The use of the trust–region strategy in the Gauss–Newton method ensures the global convergence of the algorithm. The theory of the global convergence of trust–region methods is already well established, see e. g. Powell [83], Powell [84], Thomas [102], Moré [76, Theorem (4.14)]. In [18, Section 16.3] Conn, Gould and Toint present also convergence results of trust–region methods applied to the nonlinear least–squares problem. Therefore, we give only a rough sketch on the main aspects necessary to ensure the global convergence of the algorithm presented in Chapter 2.

Firstly, we state the assumptions which have to be satisfied by the residual function $R(x)$ in the nonlinear least–squares problem (2.1.3).

Assumption 3.1.1 The function $R(x)$ is twice continuously differentiable on $I\!\!R^n$.

Assumption 3.1.2 The function $R(x) = (R_1(x), \ldots, R_m(x))^T$, its Jacobian $J(x)$ and the Hessian $H_i(x)$ of the residuals $R_i(x)$, $i = 1, \ldots, m$ are uniformly bounded, i. e. there exists a positive constant K such that for all $x \in I\!\!R^n$ and for all $i = 1, \ldots, m$ holds

$$\|R(x)\|_2 \le K, \quad \|J(x)\|_2 \le K, \quad \text{and} \quad \|H_i(x)\|_2 \le K.$$

The latter assumption is strong and will not be fulfilled in general by the residual functions modelling the systems in Chapters 4 and 5. In the proof of the global convergence, it is sufficient to assume that the Hessian $\|\nabla^2\phi(x)\|_2$, and thus $\|R(x)\|_2$, $\|J(x)\|_2$ and $\|H_i(x)\|_2$, $i = 1, \ldots, m$, are bounded for elements which lie between two successive iterates of the Inexact Gauss–Newton Method 2.2.3, see Conn, Gould, and Toint [18, Section 16.3]. Hence, if the Gauss–Newton iterates x_i stay in a compact subset of $I\!\!R^n$ the smoothness of the residual function R ensures this weaker condition automatically.

Note that the characteristics of the applications examined in Chapters 4 and 5 will guarantee this boundedness of the iterates.

In neural networks, the output is computed by a repeated evaluation of so–called activation functions. In Chapter 4, this activation function is given by the logistic function $\sigma_L(x) = 1/(1 + e^{-2x})$. For large arguments, i.e. $|x|$ is large, this function is almost constant. Thus, if large parameters are present in the neural network, its output will be almost constant independently from the input data of the neural network which is an undesirable effect. Therefore, in the context of neural networks, it is natural that the parameter vector x stays in a compact region.

In Chapter 5, we are determining the volatility parameter in an extended Black–Scholes model by approximating the theoretical computed European option prices and their market counterparts and by simultaneously minimizing the variation of the volatility parameter. Since a certain size of the volatility parameter is given by the market European option prices, the minimization of the variation of the volatility parameter guarantees that the parameter vector will not get arbitrary large. Thus, in this model, the affiliation of the Gauss–Newton iterates x_i to a compact region is also naturally given by the characteristics of the examined problem.

An additional condition, that is necessary for the convergence theory of trust–region methods, is a sufficient reduction of the model function within the trust region $\mathcal{U}_i$ (see (2.2.6) for its definition). The descent direction which reduces the quadratic model $\varphi_i(\delta x)$ in a small local neighborhood of the current iterate x_i at the fastest rate is the steepest descent direction δx_i^{sd} of $\phi(x)$. The minimum of the model along this direction within the trust region, i. e.

$$x_i^{cp} = x_i - \vartheta_i^{cp}\delta x_i^{sd} \quad \text{with} \quad \vartheta_i^{cp} = \arg \min_{\substack{\vartheta \ge 0 \\ x_i - \vartheta\delta x_i^{sd} \in \mathcal{U}_i}} \varphi(-\vartheta\delta x_i^{sd}) \tag{3.1.1}$$

which is the Cauchy point (see Definition 2.2.3), is computed easily in the case of our quadratic model. In Algorithm 2.2.3, this is just the first iterate computed by the Modified Preconditioned Conjugate Gradient Method 2.2.2 in Step 1 of the former algorithm.

This Cauchy point fulfills the following decrease condition (see Powell [84, Theorem 4] or Conn, Gould, and Toint [18, Theorem 6.3.1]).

Lemma 3.1.1 *Let the Cauchy point* x_i^{cp} *be given according to* (3.1.1) *and define*

$$\delta x_i^{cp} \;=\; -\,\vartheta_i^{cp} \delta x_i^{sd} \,.$$

Then, the model function $\varphi_i(\delta x)$ *(see* (2.2.2) *for its definition) satisfies*

$$\varphi_i(0) - \varphi_i(\delta x_i^{cp}) \;\geq\; \tfrac{1}{2} \|J(x_i)^T R(x_i)\|_2 \;\min \left[\frac{\|J(x_i)^T R(x_i)\|_2}{q_i} \,,\; \nu_i^{cp} \Delta_i \right] \tag{3.1.2}$$

where

$$q_i \;=\; 1 + \max_{x \in \mathcal{U}_i} \; \|J(x)^T J(x)\|_2 \,,$$

with the trust region $\mathcal{U}_i$ *defined in* (2.2.6)*, is an upper bound on the curvature of the model and where*

$$\nu_i^{cp} \;=\; \frac{\|J(x_i)^T R(x_i)\|_2}{\|J(x_i)^T R(x_i)\|_{\bar{D}}} \,.$$

Moreover, from Lemma 2.2.6(a) follows for each iterate δx_k^i generated in the kth iteration of the Modified Preconditioned Conjugate Gradient Method 2.2.2 in the ith iteration of the Inexact Gauss–Newton Method 2.2.3 (see Steihaug [98])

$$\varphi_i(0) - \varphi_i(\delta x_k^i) \;\geq\; \tfrac{1}{2} \; \|J(x_i)^T R(x_i)\|_2 \;\min \left[\frac{\|J(x_i)^T R(x_i)\|_2}{q_i} \,,\; \nu_i^{cp} \Delta_i \right] \tag{3.1.3}$$

with q_i and ν_i^{cp} defined as in Lemma 3.1.1. From this last condition also follows that the quotient ρ_i in Step 2 of Algorithm 2.2.3 is well–defined since (3.1.3) yields that

$$\varphi_i(\delta x_k^i) \;<\; \varphi(0)$$

as long as $J(x_i)^T R(x_i) \neq 0$ and $\delta x_k^i \neq 0$.

Then, the following theorem states the global convergence of the Inexact Gauss–Newton Method 2.2.3 presented in Chapter 2 (see for instance Conn, Gould, and Toint [18, Theorem 6.4.6]).

Theorem 3.1.2 *Let Assumption 3.1.1 and Assumption 3.1.2 be satisfied. Then, for the sequence* $\{x_i\}$ *generated by Algorithm 2.2.3 holds*

$$\lim_{i \to \infty} J(x_i)^T R(x_i) \;=\; 0 \,,$$

i. e. the limit points of the sequence $\{x_i\}$ *generated by the Inexact Gauss–Newton Algorithm 2.2.3 satisfy the first–order necessary optimality condition for the nonlinear least–squares problem* (2.1.3).

3.2 Local Convergence Analysis of the Inexact Gauss–Newton Method

In this section, we examine the local convergence of the inexact Gauss–Newton method without making a distinction between overdetermined and underdetermined residual functions.

The local convergence of the Gauss–Newton method has already been analyzed extensively, see for example [15], [11], [30], [29], [50], [25], [10], [93], [26]; however, none of these results make any statements about the q–convergence rate of the inexact Gauss–Newton method applied to underdetermined residual vectors $R(x)$.

In the following subsection, firstly, we give an overview of already existing local convergence analysis.

3.2.1 Known Local Convergence Results for the Gauss–Newton Method

We start with statements about the application of the Gauss–Newton method to the generally examined class of overdetermined residual functions $R(x)$ whose Jacobian $J(x)$ at the solution x_* has a full rank. Subsequently, we state results for an extended class of residual functions.

Brown and Dennis show in [15] that the Gauss–Newton method converges q–linearly when the residual vector $R : \mathbb{R}^n \to \mathbb{R}^m$ in the nonlinear least–squares problem is overdetermined, i.e. $m \geq n$, and the Jacobian $J(x)$ of $R(x)$ at the solution x_* has a full column rank. If additionally, the residual vector $R(x)$ is zero at the solution x_*, then this method converges even q–quadratically (see Theorem 1 – 2, Corollary 1 – 2 and Remark 2 in [15] as well as Dennis and Schnabel, Theorem 10.2.1 and Corollary 10.2.2 in [25]). The discussion by Dennis and Schnabel in [25] makes clear that the nonlinearity of the residual function and the size of the residual at the solution are important quantities in the proof of these results. They remark that the speed of convergence depends on these quantities in connection with the smallest eigenvalue of $J(x_*)^T J(x_*)$. Moreover, if either the relative nonlinearity or the relative residual size of the problem is too large, the Gauss–Newton method may not converge at all.

The convergence results of Brown and Dennis have the disadvantage that they do not consider the invariance of the Gauss–Newton sequence under unitary transformations (see e.g. Subsection 3.2.3). This aspect is considered by Häußler in [50]. He shows that for overdetermined residual functions with $\operatorname{rank}(J(x_*)) = n$ the Gauss–Newton method converges q–linearly or q–quadratically, respectively, under assumptions which are also invariant under unitary transformations (see Theorems 2.1, 2.5, and 2.7 in [50]).

Deuflhard and Apostolescu examine in [29] the Gauss–Newton method also for overdetermined residual vectors; however, they drop the assumption about the full rank of the Jacobian at the solution. They show under unitary invariant assumptions that the Gauss–Newton method converges locally for adequate nonlinear least–squares problems if the norm minimal solution of the Gauss–Newton subproblems is chosen for the correction of the Gauss–Newton iterates (see Theorem 1 in [29]). On this occasion, they call a nonlinear least–squares problem adequate, if and only if, there exists some convex neighborhood of the solution x_* so that

$$\|J(y)^+ \left(I - J(x)\, J(x)^+\right) R(x)\|_2 \;\leq\; \chi(x)\, \|y - x\|_2 \quad \text{with} \quad \chi(x) \;\leq\; \overline{\chi} \;<\; 1 \tag{3.2.1}$$

holds for all elements x, y in this neighborhood. A combination of the critical point condition of x_*, i.e. $J(x_*)^T R(x_*) = J(x_*)^+ R(x_*) = 0$, with the condition (3.2.1) yields

$$\|J(y)^+ R(x_*)\|_2 \;\leq\; \chi(x_*)\;\|y - x_*\|_2$$

such that condition (3.2.1) can be interpreted as some kind of "small residual" condition on the residual vector $R(x)$ [29]. The result of Deuflhard and Apostolescu does not contain any statements about the rate of convergence of $\|x_k - x_*\|_2$, where x_k is the kth Gauss–Newton iterate.

The local convergence analyses of the Gauss–Newton method by Boggs [11], Deuflhard and Heindl [30], and Bock [10] do not distinguish between overdetermined and underdetermined residual vectors $R(x)$. Boggs proves in [11] the local convergence of the Gauss–Newton method if the norm minimal solution of the Gauss–Newton subproblems is chosen for the update of the Gauss–Newton iterates (see Theorem 3.2 in connection with his statements about the bounds on the step size in [11]). This result is valid for residual functions whose Jacobian $J(x)$ has a constant rank and $J(x)^T R(x)$ is Lipschitz continuous in a convex neighborhood of the solution x_* and if the largest eigenvalue of $J(x_*)^T J(x_*)$ is smaller than 2. If this last assumption on the largest eigenvalue of $J(x_*)^T J(x_*)$ is not satisfied, then he obtains the local convergence for a damped Gauss–Newton method.

Deuflhard and Heindl present in [30] an unitary invariant convergence theorem for a class of generalized Gauss–Newton methods. They prove under assumptions that are invariant under unitary transformations the local convergence to a critical point x_* for this class of generalized Gauss–Newton methods (see Theorem 4 in [30]). The ordinary Gauss–Newton method where the Gauss–Newton iterates are updated by using the norm–minimal solution of the Gauss–Newton subproblems is contained in this class. The critical condition in their result is again the condition (3.2.1).

Furthermore, Bock proves in [10] the r–linear convergence of the Gauss–Newton sequence if the norm minimal solution of the Gauss–Newton subproblems is chosen for the correction of the Gauss–Newton iterate (see Theorem 3.1.44 in [10]). He introduces a generalized inverse which makes in his case an uniform treatment of constrained as well as unconstrained nonlinear least–squares problems possible. The critical condition in his result is given by

$$\left\|\left(J(y)^{(+)} - J(x)^{(+)}\right)\;Q(x)\right\|_2 \;\leq\; \chi(x)\;\|y - x\|_2 \quad \text{with} \quad \chi(x) \;\leq\; \overline{\chi} \;<\; 1$$

where $J(x)^{(+)}$ is a generalized inverse of the Jacobian $J(x)$ and $Q(x)$ is given by

$$Q(x) \;:=\; R(x) - J(x)\,J(x)^{(+)} R(x) \;=\; \left(I - J(x)\,J(x)^{(+)}\right)\;R(x)\,.$$

If unconstrained nonlinear least–squares problems are treated, the generalized inverse $J(x)^{(+)}$ of the Jacobian $J(x)$ in Bock's examination is just the Moore–Penrose inverse $J(x)^+$ so that in this case the above given condition is identical to condition (3.2.1) by Deuflhard and Heindl because of the identity $J(x)^+\,(I - J(x)\;J(x)^+) = 0$.

Moreover, Dennis and Steihaug examined in [26] the convergence rate of an inexact version of the Gauss–Newton method in the overdetermined, full rank case. Under similar assumptions as Dennis and Schnabel, they showed that the inexact Gauss–Newton iterates converge at least

q–linearly to the solution in a weighted norm (see Theorem 3.1 in [26]). The convergence of this sequence, again, depends on the nonlinearity and the size of the residual of the problem. Additionally, since the linear Gauss–Newton subproblems are only solved inexactly, the residuals of the linearized problems, i.e. $J(x_i)^T r_i = J(x_i)^T (J(x_i)\,\delta x_i + R(x_i))$, influence the convergence of the iterates to a solution.

3.2.2 Difficulties in the Convergence Analysis of the Inexact Gauss–Newton Method Applied to Underdetermined Problems

The results of the q–convergence rate of the Gauss–Newton method mentioned above cannot be applied to our problem formulation because the nonlinear least–squares problems under consideration are, in general, underdetermined.

As far as we know, there do not exist any results of the q–convergence rate of the Gauss–Newton method applied to nonlinear least–squares problems with an underdetermined residual vectors.

In this case, the Jacobian $J(x)$ of the residual function $R(x)$ has a nontrivial null space. However, this implies the difficulty that the linear least–squares problems, which have to be solved in every Gauss–Newton iteration, do not have unique solutions. The set of solutions of the linear least–squares problem

$$\text{Minimize} \quad \tfrac{1}{2}\,\|J(x_i)\,\delta x + R(x_i)\|_2^2\,, \quad \delta x \in I\!R^n \tag{3.2.2}$$

is given by

$$S_i \;=\; \{\delta x_i \in I\!R^n \mid \; \delta x_i \;= -J(x_i)^+ R(x_i) + (I - J(x_i)^+ J(x_i))\; z\;,\; z \in I\!R^n\}$$

where $J(x_i)^+$ is the Moore–Penrose inverse, also known as pseudoinverse, of $J(x_i)$. The first term in the representation of the iterates δx_i in S_i is the minimal norm solution to (3.2.2) which is orthogonal to the null space of $J(x_i)$. Since $(I - J(x_i)^+ J(x_i))$ is the orthogonal projection onto the null space of $J(x_i)$ the second term represents an arbitrary element in the null space of $J(x_i)$.

If iterative methods for the solution of the linear least–squares problem (3.2.2) are used in combination with a trust–region method, then the linear least–squares problem (3.2.2) is solved inexactly. Let δx_i be the approximate solution to (3.2.2). Then, (2.2.11) implies that the residual r_i satisfies

$$-J(x_i)^T r_i \;=\; J(x_i)^T (J(x_i)\,\delta x_i + R(x_i)) \tag{3.2.3}$$

if the Modified Preconditioned Conjugate Gradient Method 2.2.2 is used for the solution of (3.2.2). Since the linear system of equations (3.2.3) is equivalent to the linear least–squares problem

$$\text{Minimize} \quad \tfrac{1}{2}\,\|J(x_i)\,\delta x + (R(x_i) + r_i)\|_2^2\,, \quad \delta x \in I\!R^n$$

the set of solutions for the perturbed normal equations (3.2.3) is given by

$$\tilde{S}_i \;=\; \{\delta x_i \in I\!R^n \mid \; \delta x_i \;= -J(x_i)^+ (R(x_i) + r_i) + (I - J(x_i)^+ J(x_i))\; z\;,\; z \in I\!R^n\}\,. \tag{3.2.4}$$

3.2.3 Local q–Convergence Analysis of an Arbitrary Inexact Gauss–Newton Iteration

Firstly, we analyze the convergence behavior when in each Gauss–Newton iteration an arbitrary element of $\tilde{S}_i$ (3.2.4) is chosen for the correction of the Gauss–Newton iterate.

We start with proposing some conditions on the residual vector $R(x)$ and its Jacobian that will be used in the following convergence Theorem 3.2.1.

Assumption 3.2.1 Let a real constant $\varrho \geq 0$ be given with

$$\begin{aligned} &\|J(x)^+(J(x+t(x_*-x))-J(x))(x-x_*)\|_2 \\ &\qquad\qquad \leq\ \varrho\, t\, \|J(x)^+J(x)\,(x-x_*)\|_2^2\,,\ t \in [0\,,1], \end{aligned} \tag{3.2.5}$$

for all x in a set Ω.

Assumption 3.2.2 Let a real constant $\chi \geq 0$ be given with

$$\|J(x)^+J(x)\,(J(x)^+-J(x_*)^+)R(x_*)\|_2\ \leq\ \chi\,\|J(x)^+J(x)\,(x-x_*)\|_2 \tag{3.2.6}$$

for all x in a set Ω.

When in each Gauss–Newton iteration the Gauss–Newton iterate is updated by using an arbitrary element of $\tilde{S}_i$ (3.2.4), then the new iterate is an arbitrary element in an affine linear subspace, i.e. $x_{i+1} \in y_i + \mathcal{N}(J(x_i))$ with $y_i = x_i - J(x_i)^+(R(x_i) + r_i)$ and $\mathcal{N}(J(x_i))$ is the null space of $J(x_i)$. Thus, it is only possible to obtain the q–linear convergence of the iterates on the orthogonal complement of the null space of the Jacobian, i.e. on a linear subspace.

Theorem 3.2.1 *Let* $R : I\!R^n \to I\!R^m$ *be continuously differentiable on an open and convex set* $\Omega \subset I\!R^n$. *Furthermore, let the sequence*

$$x_{i+1}\ =\ x_i - J(x_i)^+R(x_i) - (I - J(x_i)^+J(x_i))\,z_i - J(x_i)^+r_i\,,\quad z_i \in I\!R^n \tag{3.2.7}$$

with

$$\|J(x_i)^+r_i\|_2\ \leq\ \eta_i\,\|J(x_i)^+J(x_i)\,(x_i-x_*)\|_2\,,\quad 0\ \leq\ \eta_i\ \leq\ \eta\ <\ 1\,, \tag{3.2.8}$$

and $x_0 \in I\!R^n$ *converge to a solution* $x_* \in \Omega$ *with* $J(x_*)^T R(x_*) = 0$. *Let the Assumptions 3.2.1 and 3.2.2 be satisfied for all* $x \in \Omega$ *with* $\chi + \eta < 1$.
Then there exists $\epsilon > 0$ *such that* $c := \frac{\varrho\epsilon}{2} + \chi + \eta < 1$ *and the sequence* $\{x_i\}$ *satisfies either* $J(x_i)^T R(x_i) = 0$ *for some finite* i *or*

$$\begin{aligned} &\|J(x_i)^+J(x_i)\,(x_{i+1}-x_*)\|_2\ \leq \\ &\qquad \tfrac{\varrho}{2}\,\|J(x_i)^+J(x_i)\,(x_i-x_*)\|_2^2 + (\chi+\eta)\,\|J(x_i)^+J(x_i)\,(x_i-x_*)\|_2 \end{aligned} \tag{3.2.9}$$

and

$$\|J(x_i)^+ J(x_i)\,(x_{i+1}-x_*)\|_2 \;\le\; c\,\|J(x_i)^+ J(x_i)\,(x_i - x_*)\|_2 \tag{3.2.10}$$

for sufficiently large i.

Proof: The proof follows by induction over i. Define $e_i := x_i - x_*$ and set $R_i := R(x_i)$, $J_i := J(x_i)\,, J_i^+ := J(x_i)^+$ and choose ϵ so small that $N(x_*\,,\epsilon) \subset \Omega$ and $\frac{\varrho\epsilon}{2} + \chi + \eta < 1$.
Because of the identity $\mathcal{N}(J(x_*)^T) = \mathcal{N}(J(x_*)^+)$, $J(x_*)^T R(x_*) = 0$ and the mean value theorem

$$\begin{aligned} e_{i+1} &= e_i - J_i^+ R_i - (I - J_i^+ J_i)\, z_i - J_i^+ r_i \\ &= (I - J_i^+ J_i)(e_i - z_i) - J_i^+ (R_i - R_* - J_i e_i) - (J_i^+ - J_*^+) R_* - J_i^+ r_i \\ &= (I - J_i^+ J_i)(e_i - z_i) - J_i^+ \int_0^1 (J(x_i + t(x_* - x_i)) - J_i)\, e_i\, dt \\ &\qquad\qquad\qquad\qquad\qquad - (J_i^+ - J_*^+) R_* - J_i^+ r_i . \end{aligned}$$

Since $(I - J_i^+ J_i)$ is the orthogonal projection onto the null space of J_i and because the Moore–Penrose inverse $J_i{}^+$ fulfils the condition $J_i^+ = J_i^+ J_i\, J_i^+$, multiplication of e_{i+1} with the projection $J_i^+ J_i$ onto the range of J_i^T yields

$$J_i^+ J_i\, e_{i+1} \;=\; -\int_0^1 J_i^+ (J(x_i + t(x_* - x_i)) - J_i)\, e_i\, dt \;-\; J_i^+ J_i\, (J_i^+ - J_*^+)\, R_* - J_i^+ r_i\,.$$

Taking the Euclidean norm of $J_i^+ J_i\, e_{i+1}$ yields

$$\begin{aligned} \|J_i^+ J_i\, e_{i+1}\|_2 \;&\le\; \int_0^1 \|J_i^+ (J(x_i + t(x_* - x_i)) - J_i)\, e_i\,\|_2\, dt \\ &\qquad\qquad + \|J_i^+ J_i\, (J_i^+ - J_*^+)\, R_*\,\|_2 + \|J_i^+ r_i\|_2\,. \end{aligned}$$

With (3.2.5), (3.2.6), and (3.2.8) we obtain

$$\begin{aligned} \|J_i^+ J_i\, e_{i+1}\|_2 \;&\le\; \tfrac{\varrho}{2}\, \|J_i^+ J_i\, e_i\|_2^2 + (\chi + \eta)\, \|J_i^+ J_i\, e_i\|_2 \\ &\le\; \left(\tfrac{\varrho}{2}\, \|J_i^+ J_i\|_2\, \|e_i\|_2 + \chi + \eta\right)\, \|J_i^+ J_i\, e_i\|_2\,. \end{aligned}$$

Since by assumption x_i converges to x_* there exists $i_0 = i_0(\epsilon)$ such that $x_i \in N(x_*, \epsilon)$ for all $i > i_0$. Because $\|J_i^+ J_i\|_2 = 1$ we have for $i > i_0$

$$\begin{aligned} \|J_i^+ J_i\, e_{i+1}\|_2 \;&\le\; \left(\tfrac{\varrho\epsilon}{2} + \chi + \eta\right)\, \|J_i^+ J_i\, e_i\|_2 \\ &=\; c\, \|J_i^+ J_i\, e_i\|_2 \end{aligned}$$

such that (3.2.9) and (3.2.10) are proved. ■

Note that the Gauss–Newton sequence (3.2.7) is invariant under unitary transformations of the residual vector $R(x)$, i.e. the sequence generated according to formula (3.2.7) is the same if it is applied to $UR(x)$, with $U \in I\!R^{m\times m}$ is a unitary matrix, or to $R(x)$. This follows from (see for instance Ben–Israel, Greville [6, Exercise 17(d)] for the validity of the equality $A^+ = (A^TA)^+A^T$ for an arbitrary matrix $A \in I\!R^{m\times n}$):

$$\begin{aligned} ((U\,R)'(x))^+ &= (U\,J(x))^+ \\ &= \left((U\,J(x))^T(U\,J(x))\right)^+ (U\,J(x))^T \\ &= \left(J(x)^TU^TU\,J(x)\right)^+ J(x)^TU^T \\ &= \left(J(x)^TJ(x)\right)^+ J(x)^TU^T \\ &= J(x)^+U^T \end{aligned}$$

for all $x \in \Omega$ and any unitary matrix $U \in I\!R^{m\times m}$ (see for instance [30]). Thus,

$$(U\,J(x))^+U\,R(x) \;=\; J(x)^+U^TU\,R(x) \;=\; J(x)^+R(x)$$

and for $z \in I\!R^n$

$$(I - (U\,J(x))^+U\,J(x))\,z \;=\; (I - J(x)^+U^TU\,J(x))\,z \;=\; (I - J(x)^+J(x))\,z\,.$$

Furthermore, for the residual $\bar{r}_i$ of the linear least–squares problem

$$\text{Minimize} \quad \tfrac{1}{2}\,\|U\,J(x_i)\,\delta x + U\,R(x_i)\|_2^2\,, \qquad \delta x \in I\!R^n$$

when solved with the modified preconditioned conjugate gradient method 2.2.2 holds

$$\bar{r}_i \;=\; U\,J(x_i)\delta x_i + U\,R(x_i) \;=\; U\;(J(x_i)\delta x_i + R(x_i)) \;=\; U\,r_i$$

such that

$$(U\,J(x_i))^+\bar{r}_i \;=\; J(x_i)^+U^TU\,r_i \;=\; J(x_i)^+r_i\,.$$

Hence, any unitary transformation of the residual vector $R(x)$ will not affect the convergence or divergence of the inexact Gauss–Newton sequence. This is reflected also by Theorem 3.2.1 since the assumptions and the statements in Theorem 3.2.1 remain unaltered, when $R(x)$ is replaced by $U\,R(x)$ for any unitary matrix $U \in I\!R^{m\times m}$. Thus, Theorem 3.2.1 is invariant under unitary transformations. This characteristic will also be valid for the other theorems and corollaries in this section.

Since the q–linear convergence of the inexact Gauss–Newton sequence (3.2.7) in Theorem 3.2.1 is obtained only on the orthogonal complement of the null space of the Jacobian, it is natural that the conditions on the residual vector $R(x)$ and the residuals r_i of the linear least–squares problems are also restricted to this linear subspace. Note that this argument is considered in the Assumptions 3.2.1, 3.2.2, and (3.2.8) by the projection of both sides of each of these inequalities onto the orthogonal complement of the null space of the Jacobian $J(x_i)$. Without these projections, Assumptions 3.2.1 and 3.2.2 are similar to those given, for instance, by Deuflhard and Heindl [30] or Bock [10], evaluated at $x = x_*$.

Furthermore, the constant χ in Assumption 3.2.2 plays a crucial role in Theorem 3.2.1 since χ must be less than 1 to guarantee convergence of the sequence (3.2.7). Using the critical point condition of x_*, i.e. $J(x_*)^T R(x_*) = J(x_*)^+ R(x_*) = 0$ and the identity $J(x)^+ = J(x)^+ J(x)\, J(x)^+$ the left side in inequality (3.2.6) reduces to

$$\begin{aligned}
&\|J(x)^+ J(x)\ (J(x)^+ - J(x_*)^+)\ R(x_*)\|_2 \\
&\qquad = \ \|J(x)^+ J(x)\, J(x)^+ R(x_*) - J(x)^+ J(x)\, J(x_*)^+ R(x_*)\|_2 \\
&\qquad = \ \|J(x)^+ R(x_*)\|_2
\end{aligned}$$

Thus, as one would expect, Theorem 3.2.1 also includes some kind of small residual condition on the residual vector $R(x)$ at the solution x_* (see also Deuflhard and Apostolescu [29]). Moreover, the following equality holds for the left side in inequality (3.2.6):

$$\begin{aligned}
\|J(x)^+ J(x)\ (J(x)^+ - J(x_*)^+)\ R(x_*)\|_2 &= \|J(x)^+ R(x_*)\|_2 \\
&= \|\,(J(x)^+ - J(x_*)^+)\ R(x_*)\|_2
\end{aligned}$$

This shows that Assumption 3.2.2 also can be interpreted as a combined measure of the continuity of the Moore–Penrose inverse of the Jacobian $J(x)$ and on the size of the residual vector $R(x)$ at the solution x_*. Hence, if the Lipschitz constant of the Moore–Penrose inverse of $J(x)$ in a convex neighborhood of the solution x_* is very small than Theorem 3.2.1 also holds for residual functions $R(x)$ with a larger size of the residual at the solution x_*. If the residual function $R(x)$ is linear, the size of the residual is even irrelevant for the convergence of the Gauss–Newton method. However, note that a necessary condition for the continuity of the Moore–Penrose inverse of the Jacobian is a constant rank of the Jacobian $J(x)$ for all x in the specific neighborhood of the solution x_* (see for instance Björck [7, Corollary 1.4.1]). Furthermore, Theorem 3.2.1 points out that the inexact Gauss–Newton sequence (3.2.7) does not converge at all if one of these quantities is too large.

Inequality (3.2.8) in Theorem 3.2.1 controls the size of the residual of the linearized Gauss–Newton subproblems. Note that (see for instance [6, Exercise 17(d)])

$$\begin{aligned}
\|J(x_i)^+ r_i\|_2 &= \|\left(J(x_i)^T J(x_i)\right)^+ J(x_i)^T r_i\|_2 \\
&\le \|\left(J(x_i)^T J(x_i)\right)^+\|_2\ \|J(x_i)^T r_i\|_2 \\
&= \|\left(J(x_i)^T J(x_i)\right)\|_2^{-1}\ \|J(x_i)^T r_i\|_2
\end{aligned}$$

where $J(x_i)^T r_i = J(x_i)^T\ (J(x_i)\,\delta x_i + R(x_i))$ is the error made in the solution of the linearized problem if for instance an iterative method is used for the solution of this problem. Therefore, condition (3.2.8) can be interpreted as a relative measure on the size of the error in the solution of the linearized problem in connection with the smallest singular value of the Jacobian $J(x_i)$.

If the nonlinear least–squares problem is a zero residual problem, i.e. $R(x_*) = 0$, it follows from Assumption 3.2.2 that the constant χ in (3.2.6) is equal zero. If, in addition, η_i satisfies a proper bound, then the following corollary is a direct consequence of Theorem 3.2.1.

Corollary 3.2.2 *Let the assumptions of Theorem 3.2.1 be satisfied. If* $R(x_*) = 0$ *and* η_i *in (3.2.8) satisfies*

$$\eta_i \;\le\; \gamma \, \| J(x_i)^+ J(x_i) \, (x_i - x_*) \, \|_2 \, ,$$

then there exists $\epsilon > 0$ *such that the sequence* $\{x_i\}$ *satisfies either* $J(x_i)^T R(x_i) = 0$ *for some finite* i *or*

$$\|J(x_i)^+ J(x_i) \, (x_{i+1} - x_*)\|_2 \;\le\; \left(\tfrac{\varrho}{2} + \gamma\right) \; \|J(x_i)^+ J(x_i) \, (x_i - x_*)\|_2^2 \, .$$

3.2.4 Local q–Convergence Analysis of the Inexact Gauss–Newton Iteration Generated by the Proposed Algorithm

It is well known that the solution generated by the conjugate gradient method is the norm minimal solution to the linear least–squares problem if only the starting element δx_0 is zero (see e.g. Hestenes [54, Section 4], [55, p. 296]). Hence, if we set $z_i = 0$ for $i \in I\!N$ in (3.2.7) then the sequence (3.2.7) of Theorem 3.2.1 corresponds to the sequence generated by the Inexact Gauss–Newton Algorithm 2.2.3. Hence, from Theorem 3.2.1 follows the next corollary.

Corollary 3.2.3 *Let the assumptions of Theorem 3.2.1 be satisfied. Furthermore let the sequence*

$$x_{i+1} = x_i - J(x_i)^+ R(x_i) - J(x_i)^+ r_i \, , \tag{3.2.11}$$

with $x_0 \in I\!R^n$ *converge to a solution* $x_* \in \Omega$ *with* $J(x_*)^T R(x_*) = 0$.
Then there exists $\epsilon > 0$ *such that* $c := \frac{\varrho \epsilon}{2} + \chi + \eta < 1$ *and the sequence* $\{x_i\}$ *satisfies either* $J(x_i)^T R(x_i) = 0$ *for some finite* i *or*

$$\begin{aligned} &\|J(x_i)^+ J(x_i) \, (x_{i+1} - x_*)\|_2 \;\le \\ &\tfrac{\varrho}{2} \, \|J(x_i)^+ J(x_i) \, (x_i - x_*)\|_2^2 \;+\; (\chi + \eta) \; \|J(x_i)^+ J(x_i) \, (x_i - x_*)\|_2 \end{aligned} \tag{3.2.12}$$

and

$$\|J(x_i)^+ J(x_i) \, (x_{i+1} - x_*)\|_2 \;\le\; c \, \|J(x_i)^+ J(x_i) \, (x_i - x_*)\|_2 \tag{3.2.13}$$

for sufficiently large i.

3.2.5 Local q–Convergence Analysis of the Inexact Gauss–Newton Iteration for Overdetermined Problems

In the case that $n \le m$ and rank $J(x_*) = n$ the Moore–Penrose inverse of $J(x_i)$ is given by

$$J(x_i)^+ \;=\; (J(x_i)^T J(x_i))^{-1} J(x_i)^T$$

for $x_i \in N(x_*, \epsilon_1)$. Then the orthogonal projection on the range space of $J(x_i)^T$ satisfies $J(x_i)^+ J(x_i) = I$, and thus, Assumptions 3.2.1 and 3.2.2 change to:

Assumption 3.2.3 Let a real constant $\varrho \geq 0$ be given with

$$\|J(x)^{+}(J(x+t(x_{*}-x))-J(x))(x-x_{*})\|_{2} \;\leq\; \varrho\, t\, \|x-x_{*}\|_{2}^{2}\,,\; t \in [0\,,1], \tag{3.2.14}$$

for all x in a set Ω.

Assumption 3.2.4 Let a real constant $\chi \geq 0$ be given with

$$\|(J(x)^{+}-J(x_{*})^{+})R(x_{*})\|_{2} \;\leq\; \chi\, \|x-x_{*}\|_{2} \tag{3.2.15}$$

for all x in a set Ω.

Hence for overdetermined problems, we obtain the following corollary:

Corollary 3.2.4 *Let $R : I\!R^{n} \to I\!R^{m}$ with $n \leq m$ be continuously differentiable on an open and convex set $\Omega \subset I\!R^{n}$. Assume that there exists a solution $x_{*} \in \Omega$ with $J(x_{*})^{T}R(x_{*}) = 0$ and rank $J(x_{*}) = n$. Let the Assumptions 3.2.3 and 3.2.4 be satisfied for all $x \in \Omega$. Assume that $\chi < 1$.*
Then there exists $\epsilon, \eta > 0$ such that $c := \frac{\varrho\epsilon}{2} + \eta + \chi < 1$ and for all $x_{0} \in N(x_{*}\,, \epsilon)$ the sequence

$$x_{i+1} \;=\; x_{i} - (J(x_{i})^{T}J(x_{i}))^{-1}J(x_{i})^{T}(R(x_{i})+r_{i})$$

with

$$\|J(x_{i})^{+}r_{i}\|_{2} \;\leq\; \eta_{i}\, \|x_{i}-x_{*}\|_{2}, \quad 0 \;\leq\; \eta_{i} \;\leq\; \eta\,,$$

either satisfies $J(x_{i})^{T}R(x_{i}) = 0$ for some finite i or converges to x_{*} and

$$\|x_{i+1}-x_{*}\|_{2} \;\leq\; \tfrac{\varrho}{2}\, \|x_{i}-x_{*}\|_{2}^{2} \;+\; (\chi+\eta)\, \|x_{i}-x_{*}\|_{2} \;\leq\; c\, \|x_{i}-x_{*}\|_{2}\,.$$

Thus, in the full rank case with $n \leq m$, Theorem 3.2.1 specializes to known results (see for instance Dennis and Schnabel [25, Theorem 10.2.1], Häußler [50, Theorem 2.1]).

Furthermore, note that in the overdetermined, full rank case Assumptions 3.2.1 and 3.2.3 are identical to the assumptions of Theorem 2.1 of Häußler in [50] which examines the q–linear convergence of the Gauss–Newton sequence with unitary invariant assumptions. Note that the convergence theorem by Dennis and Schnabel is not invariant under unitary transformations so that their conditions on the residual vector $R(x)$ are slightly different.

Moreover, Conn, Gould, and Toint show in [18, Theorem 16.1.4] that for overdetermined problems the trust–region constraint in the Inexact Gauss–Newton Algorithm 2.2.3 is asymptotically inactive and that the trust–region radius Δ_{i} is bounded away from zero if the residual vector $R(x)$ at the solution x_{*} is zero and its Jacobian $J(x_{*})$ has full column rank. Since

$$\begin{aligned} \|J(x_{i})^{+}r_{i}\|_{2} \;&=\; \|(J(x_{i})^{T}J(x_{i}))^{-1}J(x_{i})^{T}r_{i}\|_{2} \\ &\leq\; \|(J(x_{i})^{T}J(x_{i}))\|_{2}^{-1}\, \|J(x_{i})^{T}r_{i}\|_{2} \end{aligned}$$

in this case the size of the error tolerance η_{i} is determined by the relative error ξ_{i} in the Modified Preconditioned Conjugate Gradient Method 2.2.2 (see Step 9 of this algorithm) and by the smallest eigenvalue of $J(x_{i})^{T}J(x_{i})$ for i sufficiently large.

3.2.6 Local q–Convergence Analysis of a Specific Inexact Gauss–Newton Iteration

In (3.2.10), the convergence rate is established for a sequence of projected iterates. For a particular sequence $\{x_i\}_{i=0}^{\infty}$, which is generated with a specific solution $\delta x_i \in \tilde{S}_i$ of (3.2.2), the q–linear convergence of the iterates without projection to a local solution of the nonlinear least–squares problem can be shown. However, one should note that this sequence can only be chosen theoretically. For the update of the iterates we need information about the distance of the current iterate x_i to the solution x_* which is usually unknown.

Theorem 3.2.5 *Let $R : I\!R^n \to I\!R^m$ be continuously differentiable on an open and convex set $\Omega \subset I\!R^n$. Assume that there exists a solution $x_* \in \Omega$ with $J(x_*)^T R(x_*) = 0$. Let the Assumptions 3.2.3 and 3.2.4 be satisfied for all $x \in \Omega$. Assume that $\chi < 1$.*
Then, there exists $\epsilon > 0$ such that $c := \frac{\varrho\epsilon}{2} + \eta + \chi < 1$ and for all $x_0 \in N(x_*\,, \epsilon)$ the sequence

$$x_{i+1} \;=\; x_i - J(x_i)^+ R(x_i) - (I - J(x_i)^+ J(x_i))\,(x_i - x_*) - J(x_i)^+ r_i \tag{3.2.16}$$

with

$$\|J(x_i)^+ r_i\|_2 \;\le\; \eta_i\, \|x_i - x_*\|_2, \quad 0 \le \eta_i \le \eta\,, \tag{3.2.17}$$

either satisfies $J(x_i)^T R(x_i) = 0$ for some finite i or converges to x_* and

$$\|x_{i+1} - x_*\|_2 \;\le\; \tfrac{\varrho}{2}\, \|x_i - x_*\|_2^2 \;+\; (\chi + \eta)\, \|x_i - x_*\|_2 \tag{3.2.18}$$

and

$$\|x_{i+1} - x_*\|_2 \;\le\; c\, \|x_i - x_*\|_2\,. \tag{3.2.19}$$

Proof: The proof is similar to that for Theorem 3.2.1, and hence, the statements (3.2.18) and (3.2.19) are also proved by induction. Let $e_i\,, R_i\,, J_i\,, J_i^+$ be defined like in the proof of Theorem 3.2.1 and choose ϵ so small that $N(x_*\,, \epsilon) \subset \Omega$ and $\frac{\varrho\epsilon}{2} + \eta + \chi < 1$.
Because of the identity $\mathcal{N}(J_*^T) = \mathcal{N}(J_*^+)$ and $J(x_*)^T R(x_*) = 0$

$$\begin{aligned} e_{i+1} &= e_i - J_i^+ R_i - (I - J_i^+ J_i) e_i - J_i^+ r_i \\ &= -J_i^+ (R_i - R_* - J_i e_i) - (J_i^+ - J_*^+) R_* - J_i^+ r_i\,. \end{aligned}$$

Applying the mean value theorem and taking the Euclidean norm of e_{i+1}, we obtain

$$\|e_{i+1}\|_2 \;\le\; \int_0^1 \|J_i^+ (J(x_i + t(x_* - x_i)) - J_i)\, e_i\|_2\, dt + \|(J_i^+ - J_*^+)\, R_*\|_2 + \|J_i^+ r_i\|_2\,.$$

With (3.2.14), (3.2.15), and (3.2.17), we obtain

$$\begin{aligned} \|e_{i+1}\|_2 &\le \varrho\, \|e_i\|_2^2 \int_0^1 t\, dt + \chi\, \|e_i\|_2 + \eta\, \|e_i\|_2 \\ &= \tfrac{\varrho}{2}\, \|e_i\|_2^2 \;+\; (\chi + \eta)\, \|e_i\|_2\,, \end{aligned}$$

which proves (3.2.18). Since $\|e_i\|_2 \leq c^i \, \|e_0\|_2 < \epsilon$, we have

$$\|e_{i+1}\|_2 \;\; \leq \;\; (\tfrac{\varrho\epsilon}{2} + \chi + \eta) \, \|e_i\|_2 \; = \; c \, \|e_i\|_2 \, ,$$

such that (3.2.19) also holds. ■

In the case of a zero residual problem, i.e. $R(x_*) = 0$, we again obtain the q–quadratic convergence of the sequence (3.2.16) to the solution x_* if η_i for $i \in I\!N$ is sufficiently small.

Chapter 4

Forecasting with Neural Networks

After we have presented the algorithm and its convergence behavior, in the following two chapters we apply the Inexact Gauss–Newton Algorithm 2.2.3 on nonlinear least–squares problems arising from two different applications. In this chapter, we apply the inexact Gauss–Newton method on the nonlinear least–squares problem resulting from the formulation of the training process of neural networks. Furthermore, we examine the application of neural networks in financial forecasting.

We start this chapter by describing the structure of neural networks and their training process in Section 4.1. In the following Section 4.2, we present subroutines for the evaluation of the Jacobian and its transpose multiplied with a vector without storing the Jacobian explicitly. These two evaluations are necessary in the modified conjugate gradient method which is used for the solution of the Gauss–Newton subproblems in Algorithm 2.2.3. The solution of the nonlinear least–squares problem corresponds to the training process of the neural network which carries out the adaption of the neural network to the examined system. An often mentioned deficiency of neural networks is that the knowledge of neural networks of the examined system can not be extracted. However, in Section 4.3, we show how the sensitivity of the output of the neural network can be determined concerning small changes of the different input data. This gives at least some information on the influence of each of the input data on the output of the neural network. Finally, in Section 4.4, the application of neural networks in forecasting the German stock index DAX are discussed together with some numerical results.

4.1 Description of Neural Networks

Neural networks are information processing systems which orientate themselves by the structure and the functioning of the human brain. However, we do not claim to reproduce the brain but the natural model is to serve as a reference point, i.e. the special ability of learning capacity as well as the capability of pattern recognition should be obtained [19]. Neural networks are composed of a large number of uniform processing units. These processing units are the basic elements in a neural network and are called the neurons.

In a neuron the information are processed according the following procedure, see also Figure 4.1. The incoming information (these are the inputs in Figure 4.1) into a neuron are combined

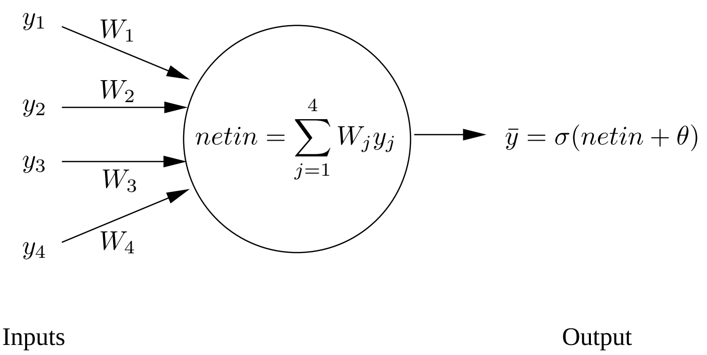

y_1
W_1
y_2
W_2
y_3
W_3
y_4
W_4
$netin = \sum_{j=1}^{4} W_j y_j$
$\bar{y} = \sigma(netin + \theta)$
Inputs
Output

FIGURE 4.1: The structure of a neuron

by weighting each of the inputs and, subsequently, adding them up, i.e. building the value $netin$ in Figure 4.1. The decision whether this neuron gives a signal to the other neurons is based on a comparison of the linear combination of the inputs with the bias, also known as the threshold, of the neuron (which corresponds to the value θ in Figure 4.1). If the sum of the linear combination of the inputs $netin$ and the bias θ is larger than zero the neuron sends a signal to the other neurons, in the opposite case it does not. This decision process of the neuron is modelled by a sigmoidal function [109].

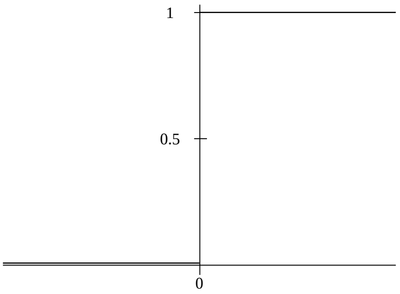


FIGURE 4.2: Heaviside function $\sigma_H(netin + \theta)$

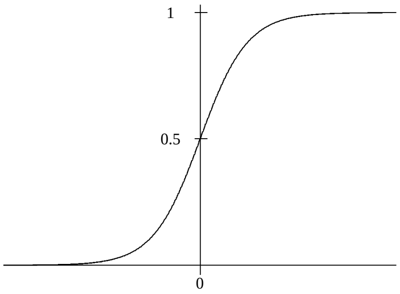


FIGURE 4.3: Logistic function $\sigma_L(netin + \theta)$ with $k = 1$ (see (4.1.1))

A yes/no statement could be described by the Heaviside function (the unit step function), see Figure 4.2,

$$\sigma_H(r) = \begin{cases} 1 & : \quad r \geq 0 \\ 0 & : \quad r < 0 \end{cases}$$

However, the Heaviside function is not differentiable at $r = 0$ so we approximate this function by the differentiable logistic function σ_L, see (4.1.1). The function that models the decision process of the neuron, is also called the transfer or activation function. Hence, the output of the neuron is generated by transmitting the sum of the linear combination of the inputs and the bias, i.e. $(netin+\theta)$, through the transfer or activation function σ_L. The output or response of the neuron is known as the activation value, or in short, the activation of the neuron [39].

In a neural network these neurons are arranged in different layers (see Figure 4.4). The first layer serves as the input layer which is fed with some given information. The neurons in this layer (which are also called the source nodes, see [51]) only make the different information available to the neural network but do not yet process the information. This takes place in the following layers, the hidden layers of the neural network. The neurons in these layers process the information according to the procedure described in the previous paragraph. Thus, only the insertion of the hidden layers enables one to represent interactions between the inputs; and, hence, allows the network to also model more complicated systems [39]. A neural network without a hidden layer is only able to map linear relations of the values in the input layer [39]. The neurons in the final layer, the output layer, of the neural network process the information for a last time; and, finally, generate the outputs of the network.

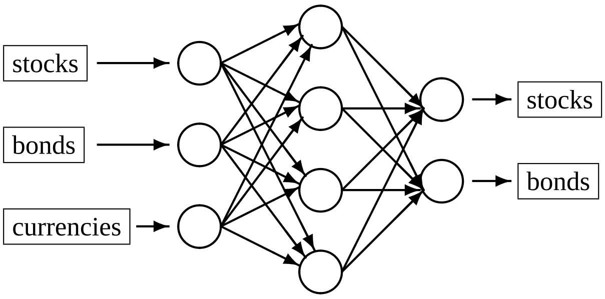


FIGURE 4.4: 3–4–2–feed–forward net

There exists a variety of different types of neural network architectures. For an overview see for instance [78, Section 17.3]. In our examinations, we restrict our attention to feed–forward networks (see Figure 4.4). These are networks which propagate the information through the network in a forward direction, on a layer–by–layer basis [51]. Thus, in these networks only the neurons in consecutive layers are connected. Finally, we suppose that the networks are fully connected, i.e. a neuron in any layer of the network is connected to all the nodes/neurons in the previous layer [51].

In order to define a neural network in mathematical terms we introduce the following notation:

$l+1$	number of layers,
n_k	number of neurons in layer k,
θ_i^k	bias or threshold of neuron i in layer k,
y_i^k	activation value of neuron i in layer k,
W_{ij}^k	weight from neuron j in layer $k-1$ to neuron i in layer k.

In the following we set

$$\begin{aligned}
\theta^k &= \left(\theta_1^k, \ldots, \theta_{n_k}^k\right)^T, & k &= 1, \ldots, l, \\
y^k &= \left(y_1^k, \ldots, y_{n_k}^k\right)^T, & k &= 0, \ldots, l, \\
W^k &= \left(W_1^k, \ldots, W_{n_{k-1}}^k\right), & k &= 1, \ldots, l,
\end{aligned}$$

with $W_j^k = (W_{1,j}^k, \ldots, W_{n_k,j}^k)^T$, $j = 1, \ldots, n_{k-1}$, such that these variables have the following dimensions

$$\begin{aligned}
\theta^k &\in \mathbb{R}^{n_k}, & k &= 1, \ldots, l, \\
y^k &\in \mathbb{R}^{n_k}, & k &= 0, \ldots, l, \\
W^k &\in \mathbb{R}^{n_k \times n_{k-1}}, & k &= 1, \ldots, l.
\end{aligned}$$

Let $\sigma_L : \mathbb{R} \to [0,1]$ be the activation function modelling the decision process of each of the neurons, e.g. the logistic function

$$\sigma_L(r) = \frac{1}{1 + e^{-kr}}. \tag{4.1.1}$$

Then the activation value y_i^k, which depends on the weights of the connections to the lower layer $k-1$, the activation values of the neurons in the previous layer $k-1$ and the bias θ_i^k, is computed by

$$y_i^k = \sigma_L\Big(\sum_{j=1}^{n_{k-1}} W_{ij}^k \, y_j^{k-1} \; + \; \theta_i^k\Big), \qquad i = 1, \ldots, n_k, \;\; k = 1, \ldots, l.$$

Using matrix notation this can be rewritten in a more compact form with

$$\sigma : \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}, \qquad \sigma(x) := (\sigma_L(x_1), \ldots, \sigma_L(x_{n_k}))^T$$

as follows

$$y^k = \sigma(W^k \, y^{k-1} \; + \; \theta^k), \qquad k = 1, \ldots, l. \tag{4.1.2}$$

Note that the y^k are defined recursively

$$\begin{aligned}
y^k &= \sigma(W^k y^{k-1} + \theta^k) = \sigma(W^k \, \sigma(W^{k-1} y^{k-2} + \theta^{k-1}) + \theta^k) \\
&= \sigma(W^k \, \sigma(W^{k-1} \ldots \sigma(W^1 y^0 + \theta^1) \ldots + \theta^{k-1}) + \theta^k).
\end{aligned} \tag{4.1.3}$$

Thus, a neural network can also be viewed as a discretized dynamical system where the initial value y^0 is given.

The representation of the activation of the neurons in (4.1.3) shows that the output of the neural network, i.e. the activation value of the neurons in layer l, depends on the weight matrices $W^1, \ldots, W^l$, the bias vectors $\theta^1, \ldots, \theta^l$, and the initial value y^0, which we express in the following by using the notation $y^l = y^l(w; y^0)$ with w defined as in (4.1.5). Thus, the knowledge of a neural network is distributed among the weight matrices $W^1, \ldots, W^l$ and the bias vectors $\theta^1, \ldots, \theta^l$ [19]. Therefore, the adaption of a neural network to a certain problem structure is carried out by adjusting these weights and biases. This will be done by presenting the network appropriate examples of input–output patterns of the examined system. Let various inputs t_p and corresponding outputs s_p,

$p = 1, \dots, P$, be given. Then, in order to learn to emulate the behavior of the system the weights W^k and θ^k, $k = 1, \dots, l$, have to be determined, depending on these inputs t_p in such a way that the output of the neural network $y^l(w; t_p)$ of (4.1.3) is close to the prescribed outputs s_p, $p = 1, \dots, P$. The closeness is described by a cost function, for example, the squared error of the neural network output with regard to the target output [39]. During the adaption of the weights, the neural network finds conformities in the presented data and maps them into its weight structure. Thus, the neural network is able to implicitly map also unknown linear and nonlinear cause/effect relations such that they are predestinated for problem classes where explicit modelling is not possible [19].

The process of the adaption of the weights is called the training of the neural network and leads to the following optimization problem

Optimization Problem (NN)

Given patterns $t_p \in I\!R^{n_0},\ s_p \in I\!R^{n_l},\ p = 1, \dots, P$.

Find $w = (W^1, \dots, W^l, \theta^1, \dots, \theta^l)$, such that

$$\sum_{p=1}^{P} \frac{1}{2} \, \|y^l(w; t_p) - s_p\|_2^2 \text{ is minimized.}$$

This is a nonlinear least squares problem with a total of

$$n_0 n_1 + n_1 n_2 + \dots + n_{l-1} n_l + n_1 + \dots + n_l$$

variables. Defining the map (cf. (4.1.6) for n and m)

$$R : I\!R^n \to I\!R^m, \quad R(w) = y^l(w) - s \tag{4.1.4}$$

with

$$\begin{aligned} y^l(w) &= (y^l(w; t_1), \dots, y^l(w; t_P)) \in I\!R^m, \\ s &= (s_1, \dots, s_P) \in I\!R^m, \\ w &= (W^1, \dots, W^l, \theta^1, \dots, \theta^l) \in I\!R^n, \end{aligned} \tag{4.1.5}$$

and

$$m = P\, n_l, \quad n = n_0 n_1 + n_1 n_2 + \dots + n_{l-1} n_l + n_1 + \dots + n_l, \tag{4.1.6}$$

we can use the Euclidean norm in $I\!R^m$ to rewrite the problem in a more compact way as

$$\text{Minimize} \quad \phi(w) = \frac{1}{2} \|R(w)\|_2^2, \quad w \in I\!R^n. \tag{4.1.7}$$

When neural networks are applied to problems in finance, like forecasting stock or index values, usually the number of patterns (t_p, s_p) available to train the network is much smaller than would be needed to specify the associated dynamical system (e.g. Zimmermann [109]). Thus the number of unknown weights and bias is often much larger than the number of training patterns, which means that $n \gg m$.

For the adaption of the weights, i.e. for the solution of the optimization problem (4.1.7), a so–called “learning algorithm” is used. The classical algorithm for training feed–forward neural networks is the backpropagation algorithm which was popularized through the publication by Rumelhart, Hinton and Williams [89]. This is a gradient–type method, for which convergence is guaranteed by applying appropriate rules for the computation of the step size along the negative gradient direction, see e.g. Grippo [47], Mangasarian and Solodov [71]. In this research we use a different approach, and that is the Inexact Gauss–Newton Method 2.2.3 presented in Chapter 2 (see Subsection 4.4.4 for a comparison of these two algorithms). In this algorithm in every iteration, we solve the constrained linear least–squares problems

$$\text{Minimize} \quad \tfrac{1}{2}\, \|R(w) + J(w)\,\delta w\|_2^2 \quad \text{subject to} \quad \|\delta w\|_2 \le \Delta$$

with the Modified Conjugate Gradient Method 2.2.2 which requires the evaluation of $J(w)\,\delta w$ as well as $J(w)^T \delta y$. In the next section, we show how both of these matrix–vector products can be computed without knowing $J(w)$ explicitly.

4.2 Evaluation of the Jacobian

We start this section with showing how the evaluation of $J(w)\,\delta w$ can be achieved without saving the matrix $J(w)$.

The main aspects in the derivation of this subroutine is the recursive definition of the activation values y^k, $k = 1, \dots, l$, in (4.1.3) and the use of the implicit function theorem. Let the function $e(y, w)$ be defined by

$$e(y,w) = (e_1^1(y,w),\, \dots, e_P^1(y,w),\, \dots, e_1^l(y,w),\, \dots, e_P^l(y,w))^T$$

with

$$\begin{aligned} y &= (y^1, \dots, y^l) \in I\!R^{\tilde m}\,, \\ y^k &= (y_1^k \;\dots, y_P^k) \in I\!R^{P n_k}\,, \\ w &= (W^1, \dots, W^l, \theta^1, \dots, \theta^l) \in I\!R^n\,, \end{aligned} \tag{4.2.1}$$

y_p^k is the activation of the neurons in layer k with input data t_p, and

$$\begin{aligned} e_p^k(y,w) &= y_p^k - \sigma(W^k\, y_p^{k-1} \;+\; \theta^k), \;\; k = 2, \dots, l, \;\; p = 1, \dots, P\,, \\ e_p^1(y,w) &= y_p^1 - \sigma(W^1\, t_p \;+\; \theta^1), \quad p = 1, \dots, P\,, \end{aligned}$$

where

$$\tilde m = P(n_1 + \ldots + n_l), \quad n = n_0 n_1 + n_1 n_2 + \ldots + n_{l-1} n_l + n_1 + \ldots + n_l\,. \tag{4.2.2}$$

In order to apply the implicit function theorem to $e(y,w) = 0$, we have to ensure that $e : I\!R^{n+\tilde m} \to I\!R^{\tilde m}$ is continuously differentiable on $I\!R^{n+\tilde m}$ and that for each $w \in I\!R^n$ there exists an unique $y \in I\!R^{\tilde{m}}$ with $e(y,w) = 0$. Furthermore, $e_y(y,w)$ has to be invertible for all $(y,w) \in I\!R^{n+\tilde{m}}$.

In our case the continuous differentiability of $e(y,w)$ follows from the continuous differentiability of σ_L and the unique solvability in y for a given w is trivial because of the iterative process (4.1.3) for computing y. The invertibility of e_y can be seen as follows. Assume that for some $\eta \in I\!R^{\tilde{m}}$ we have $e_y(y,w)\,\eta = 0$. This means

$$\eta_p^k - \sigma'(W^k y_p^{k-1} + \theta^k) W^k \eta_p^{k-1} = 0, \quad k = 2, \ldots, l\,, \quad \text{and} \quad \eta_p^1 = 0\,, \quad p = 1, \ldots, P\,.$$

From these equations we deduce $\eta = 0$. Since the assumptions of the implicit function theorem are satisfied by $e(y,w)$, it follows that there exists an unique continuous mapping

$$y : I\!R^n \to I\!R^{\tilde{m}}$$

which is defined by

$$e(y(w), w) = 0\,, \qquad \text{for all} \quad w \in I\!R^n\,,$$

and, furthermore, $y(\cdot)$ is continuously differentiable and its derivative is given by

$$y'(w) = -e_y(y(w), w)^{-1} e_w(y(w), w)\,, \qquad \text{for all} \quad w \in I\!R^n. \tag{4.2.3}$$

Note, that the last Pn_l rows of $y'(w)$ are just $J(w)$, i.e. $J(w) = (y^l)'(w)$, since s_p, $p = 1, \ldots, P$, in the definition of $R(w)$ (see (4.1.4)) do not depend on w.

Although the number of variables is very large, each component e_p^k of e depends only on a rather small number of variables, i.e. the Jacobian of e is sparse. Especially, the partial derivative of e with respect to y has a nice structure. Since $e_p^k(y,w)$ only depends on y_p^{k-1} and y_p^{k-1} the derivative $e_y(y(w), w)$ is a lower triangular matrix with identity matrices on the diagonal. Hence for obtaining the derivative of the activation of the neurons in the last layer with respect to w we just have to solve a triangular system of linear equations.

This aspect leads to the recursive computation of the matrix vector product $J(w)\,\delta w$.

Lemma 4.2.1 *Let* $w = (W^1, \ldots, W^l, \theta^1, \ldots, \theta^l)$, $\delta w = (\delta W^1, \ldots, \delta W^l, \delta\theta^1, \ldots, \delta\theta^l) \in I\!R^n$ *and*

$$y_p^k = y^k(w; t_p) = \sigma(W^k\, \sigma(W^{k-1}\, \ldots\, \sigma(W^1 t_p + \theta^1)\, \ldots + \theta^{k-1}) + \theta^k)\,, \quad p = 1, \ldots, P\,,$$

with $t_p = y_p^0$. *Set*

$$v_p^k := \sum_{j=1}^{k} \left(\frac{\partial y_p^k}{\partial W^j} \delta W^j + \frac{\partial y_p^k}{\partial \theta^j} \delta\theta^j \right) \in I\!R^{n_k}, \;\; k = 1, \ldots, l\,.$$

Then

$$J(w)\,\delta w \;=\; v^l \;=\; \left(v_1^l, \ldots, v_P^l\right)^T \in I\!R^{Pn_l} \tag{4.2.4}$$

can be computed recursively by

$$v_p^{k+1} \;=\; \sigma'(W^{k+1}y_p^k + \theta^{k+1}) \left(\delta W^{k+1}y_p^k + \delta\theta^{k+1} + W^{k+1}v_p^k\right) \tag{4.2.5}$$

for $k = 1, \ldots, l-1$ *and* $p = 1, \ldots, P$*, where*

$$v_p^1 \;=\; \sigma'(W^1 t_p + \theta^1) \left(\delta W^1 t_p + \delta\theta^1\right) \,, \quad p = 1, \ldots, P.$$

Proof: The proof follows by induction over k. First we derive the partial derivatives of y^k. Since y^k depends only on y^{k-1} and $w^k = (W^k, \theta^k)$, an application of the chain rule yields

$$\frac{\partial y_p^l}{\partial w^k} \;=\; \frac{\partial y_p^l}{\partial y_p^{l-1}} \cdots \frac{\partial y_p^{k+1}}{\partial y_p^k} \cdot \frac{\partial y_p^k}{\partial w^k}.$$

Furthermore for $\delta W^k \in I\!R^{n_k \times n_{k-1}}$

$$\begin{aligned}
\frac{\partial y_p^k}{\partial y_p^{k-1}} \;&=\; \sigma'(W^k y_p^{k-1} + \theta^k) W^k && \in \; I\!R^{n_k \times n_{k-1}} \,, \\
\frac{\partial y_p^k}{\partial W^k}\,\delta W^k \;&=\; \sigma'(W^k y_p^{k-1} + \theta^k)\,\delta W^k\, y_p^{k-1} && \in \; I\!R^{n_k} \,, \\
\frac{\partial y_p^k}{\partial \theta^k} \;&=\; \sigma'(W^k y_p^{k-1} + \theta^k) && \in \; I\!R^{n_k \times n_k} \,.
\end{aligned}$$

For $k = 1, \ldots, l$ with $y_p^0 = t_p$ we obtain

$$\begin{aligned}
\frac{\partial y_p^l}{\partial \theta^k} \;&=\; \sigma'(W^l y_p^{l-1} + \theta^l)\, W^l \;\cdots\; \sigma'(W^{k+1} y_p^k + \theta^{k+1})\, W^{k+1} \sigma'(W^k y_p^{k-1} + \theta^k) \,, \\
\frac{\partial y_p^l}{\partial W^k}\,\delta W^k \;&=\; \sigma'(W^l y_p^{l-1} + \theta^l)\, W^l \;\cdots\; \sigma'(W^{k+1} y_p^k + \theta^{k+1})\, W^{k+1} \cdot \\
&\quad\; \cdot\, \sigma'(W^k y_p^{k-1} + \theta^k)\,\delta W^k\, y_p^{k-1} \,.
\end{aligned}$$

For the first step of the induction ($k = 1$), we obtain from the definition (4.1.3)

$$\begin{aligned}
v_p^1 \;&=\; \frac{\partial y_p^1}{\partial W^1}\,\delta W^1 + \frac{\partial y_p^1}{\partial \theta^1}\,\delta\theta^1 \\
&=\; \sigma'(W^1 t_p + \theta^1) \left(\delta W^1 t_p + \delta\theta^1\right) .
\end{aligned}$$

To carry out one step of the induction argument note that $v_p^{k+1},\ k = 1, \ldots, l-1$, can be rewritten as follows

$$\begin{aligned}
v_p^{k+1} &= \sum_{j=1}^{k+1} \left(\frac{\partial y_p^{k+1}}{\partial W^j} \delta W^j + \frac{\partial y_p^{k+1}}{\partial \theta^j} \delta \theta^j \right) \\
&= \sum_{j=1}^{k+1} \sigma'(W^{k+1} y_p^k + \theta^{k+1})\, W^{k+1} \cdots \sigma'(W^{j+1} y_p^j + \theta^{j+1})\, W^{j+1} \cdot \\
&\qquad \cdot \sigma'(W^j y_p^{j-1} + \theta^j) \left(\delta W^j y_p^{j-1} + \delta \theta^j \right) \\
&= \sigma'(W^{k+1} y_p^k + \theta^{k+1}) \left(\delta W^{k+1} y_p^k + \delta \theta^{k+1} \right) + \sigma'(W^{k+1} y_p^k + \theta^{k+1})\, W^{k+1} \cdot \\
&\qquad \cdot \sum_{j=1}^{k} \sigma'(W^k y_p^{k-1} + \theta^k)\, W^k \cdots \sigma'(W^{j+1} y_p^j + \theta^{j+1})\, W^{j+1} \cdot \\
&\qquad\qquad \cdot \sigma'(W^j y_p^{j-1} + \theta^j) \left(\delta W^j y_p^{j-1} + \delta \theta^j \right) \\
&= \sigma'(W^{k+1} y_p^k + \theta^{k+1}) \left(\delta W^{k+1} y_p^k + \delta \theta^{k+1} + W^{k+1} v_p^k \right) .
\end{aligned}$$

■

For the computation of v_p^{k+1} are $3\, n_{k+1} n_k$ multiplications and n_{k+1} evaluations of σ'_L, $k = 1, \ldots, l-1$, $p = 1, \ldots, P$, necessary. Furthermore, the computation of v_p^1 requires $2\, n_1 n_0$ multiplications and n_1 evaluations of σ'_L, $p = 1, \ldots, P$. Thus, the computational effort of one evaluation of $J(w)\, \delta w$ is

$$\begin{array}{ll}
\text{\# multiplications} & P(2 n_0 n_1 + 3(n_1 n_2 + \ldots + n_{l-1} n_l)) \\
\text{\# evaluations of } \sigma'_L & P(n_1 + \ldots + n_l)
\end{array} \tag{4.2.6}$$

Note, that we have not considered the effort for the recursive computation of y_p^k in this list, yet.

The following lemma shows how the evaluation of $J(w)^T \delta y$ can be done without storing the matrix $J(w)^T$. This result follows also from the representation of $y'(w)$ in (4.2.3) and from Lemma 4.2.1. From (4.2.3) follows that

$$y'(w)\, \delta w = -e_y(y(w), w)^{-1} e_w(y(w), w)\, \delta w \, .$$

Thus,

$$\begin{aligned}
\left(J(w)^T \delta y \right)^T \delta w &= \left((y^l)'(w)^T \delta y \right)^T \delta w \\
&= \left(y'(w)^T\, \overline{\delta y} \right)^T \delta w \\
&= \overline{\delta y}^T\, y'(w)\, \delta w \\
&= -\overline{\delta y}^T\, e_y(y(w), w)^{-1}\, e_w(y(w), w)\, \delta w
\end{aligned}$$

with $\overline{\delta y} = (0, \ldots, 0, \delta y)^T$ and

$$y'(w)^T\, \overline{\delta y} = -e_w(y(w), w)^T\, \mu \, , \quad \text{where} \quad e_y(y(w), w)^T\, \mu = \overline{\delta y} \, .$$

Since $e_y(y(w), w)^T$ is an upper triangular matrix, the linear system $e_y(y(w), w)^T\, \mu = \overline{\delta y}$ is solved backwards for μ.

Lemma 4.2.2 *Let* $w = (W^1, \dots, W^l, \theta^1, \dots, \theta^l) \in \mathbb{R}^n$ *and* $\delta y = (\delta y_1, \dots, \delta y_P) \in \mathbb{R}^{Pn_l}$*. Set for* $p = 1, \dots, P$

$$\begin{aligned} \mu_p^l &= \delta y_p \in \mathbb{R}^{n_l} \\ \mu_p^k &= (W^{k+1})^T \sigma'(W^{k+1} y_p^k + \theta^{k+1}) \mu_p^{k+1} \in \mathbb{R}^{n_k}\,, \quad k = l-1, l-2, \dots, 1. \end{aligned} \tag{4.2.7}$$

Then

$$J(w)^T \delta y = \bar{w} = (\bar{W}^1, \dots, \bar{W}^l, \bar{\theta}^1, \dots, \bar{\theta}^l) \in \mathbb{R}^n \tag{4.2.8}$$

can be computed by

$$\begin{aligned} \bar{\theta}^k &= \sum_{p=1}^P \sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k\,, \quad k = 1, \dots, l \\ \bar{W}^k &= \sum_{p=1}^P \sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k (y_p^{k-1})^T\,, \quad k = 1, \dots, l \end{aligned} \tag{4.2.9}$$

with $y_p^0 = t_p$*.*

Proof: For arbitrary $\delta w \in \mathbb{R}^n$, we have according to Lemma 4.2.1 and (4.2.4), (4.2.7), (4.2.8) with $\mu^l = (\mu_1^l, \dots, \mu_P^l) \in \mathbb{R}^{Pn_l}$

$$\bar{w}^T \delta w = \delta y^T J(w) \delta w = \delta y^T v^l = (\mu^l)^T v^l \tag{4.2.10}$$

For $k = l, \dots, 2$ we obtain with (4.2.7) and Lemma 4.2.1

$$\begin{aligned} (\mu^k)^T v^k &= \sum_{p=1}^P \Big((\sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k)^T (\delta W^k y_p^{k-1} + \delta\theta^k) \\ &\qquad + ((W^k)^T \sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k)^T v_p^{k-1} \Big) \\ &= (\mu^{k-1})^T v^{k-1} + \sum_{p=1}^P ((\sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k)^T (\delta W^k y_p^{k-1} + \delta\theta^k)) \end{aligned} \tag{4.2.11}$$

$$\begin{aligned} (\mu^1)^T v^1 &= \sum_{p=1}^P (\mu_p^1)^T \; (\sigma'(W^1 t_p + \theta^1)(\delta W^1 t_p + \delta\theta^1)) \\ &= \sum_{p=1}^P \big((\sigma'(W^1 t_p + \theta^1) \mu_p^1)^T \delta W^1 t_p + (\sigma'(W^1 t_p + \theta^1) \mu_p^1)^T \delta\theta^1 \big)\,. \end{aligned}$$

Using (4.2.10) and substituting recursively into (4.2.11) yields

$$\begin{aligned} \bar{w}^T \delta w &= (\mu^l)^T v^l \\ &= \sum_{k=2}^l \sum_{p=1}^P \big((\sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k)^T \delta W^k y_p^{k-1} \\ &\qquad + (\sigma'(W^k y_p^{k-1} + \theta^k) \mu_p^k)^T \delta\theta^k \big) \\ &\quad + \sum_{p=1}^P ((\sigma'(W^1 t_p + \theta^1) \mu_p^1)^T \delta W^1 t_p + (\sigma'(W^1 t_p + \theta^1) \mu_p^1)^T \delta\theta^1). \end{aligned}$$

Because of $\delta w = (\delta W^1, \ldots, \delta W^l, \delta\theta^1, \ldots, \delta\theta^l)$ and $y_p^0 = t_p$ we obtain

$$\begin{aligned}
\bar{\theta}^k &= \sum_{p=1}^{P} \sigma'(W^k y_p^{k-1} + \theta^k)\mu_p^k\,, \quad k = 1, \ldots, l,\\
\bar{W}^k &= \sum_{p=1}^{P} \sigma'(W^k y_p^{k-1} + \theta^k)\mu_p^k (y_p^{k-1})^T\,, \quad k = 1, \ldots, l.
\end{aligned}$$

The last equation follows from

$$\begin{aligned}
(\bar{W}^k)^T \delta W^k &= \sum_{p=1}^{P} (\sigma'(W^k y_p^{k-1} + \theta^k)\mu_p^k)^T \delta W^k y_p^{k-1}\\
&= \sum_{p=1}^{P} tr\,(y_p^{k-1}(\sigma'(W^k y_p^{k-1} + \theta^k)\mu_p^k)^T \delta W^k)\\
&= \sum_{p=1}^{P} \sum_{i=1}^{n_{k-1}} \sum_{j=1}^{n_k} (y_p^{k-1})_i ((\sigma'(W^k y_p^{k-1} + \theta^k)\mu_p^k)^T)_j (\delta W^k)_{j,i},
\end{aligned}$$

which completes the proof. ■

The steps of the subroutine for computing $J(w)^T \delta y$ are very similar to the computation of the gradient by the backpropagation algorithm which is essentially the reverse differentiation technique in automatic differentiation, see Saarinen, Bramely, and Cybenko [90].

For the computation of μ_k^p in (4.2.7) are $2n_{k+1}n_k + n_{k+1}$ multiplications and n_{k+1} evaluations of σ'_L necessary, $k = 1, \ldots, l-1$, $p = 1, \ldots, P$. Furthermore, the computation of one part of the sum of $\bar{W}^k$ requires $2n_k n_{k-1} + n_k$ multiplications and n_k evaluations of σ'_L. Since the sum is built over all the training patterns, the computational effort for the determination of $\bar{W}^k$ is $P(2n_k n_{k-1} + n_k)$ multiplications and Pn_k evaluations of σ'_L, $k = 1, \ldots, l$. Note that the determination of $\bar{\theta}^k$ does not require further multiplications or evaluations of σ'_L since the terms in the sum for $\bar{\theta}^k$ also have to be calculated for the determination of $\bar{W}^k$. Thus, the effort for one computation of $J(w)^T \delta y$ consists of

$$\begin{array}{ll}
\text{\# multiplications} & P\,(2n_0 n_1 + 4(n_1 n_2 + \ldots + n_{l-1} n_l)\\
 & \qquad + n_1 + 2(n_2 + \ldots + n_l))\\
\text{\# evaluations of } \sigma'_L & P(n_1 + 2(n_2 + \ldots + n_l))
\end{array} \tag{4.2.12}$$

Moreover, for the evaluation of $J(w)\,\delta w$ as well as $J(w)^T \delta y$ it is necessary to determine the activation y_p^k in (4.1.3). The effort for this computation is

$$\begin{array}{ll}
\text{\# multiplications} & P\,(n_0 n_1 + \ldots + n_{l-1} n_l)\\
\text{\# evaluations of } \sigma_L & P(n_1 + \ldots + n_l)
\end{array} \tag{4.2.13}$$

Additionally to the evaluation of $J(w)\,\delta w$ and $J(w)^T \delta y$ in each iteration of the conjugate gradient method

$$3n + 2m \;=\; 3\,(n_0 n_1 + \ldots + n_{l-1} n_l + n_1 + \ldots + n_l) + 2Pn_l$$

multiplications have to be performed such that the overall computational effort for one conjugate gradient iteration when the inexact Gauss–Newton method is applied to the training problem of a neural network amounts to

$$\begin{array}{ll} \text{\# multiplications} & (5P+3)n_0n_1 + (8P+3)(n_1n_2 + \ldots + n_{l-1}n_l) \\ & +(P+3)n_1 + (2P+3)(n_2 + \ldots + n_{l-1}) + (4P+3)n_l \\ \text{\# evaluations of } \sigma_L & P(n_1 + \ldots + n_l) \\ \text{\# evaluations of } \sigma_L' & P(2n_1 + 3(n_2 + \ldots + n_l)) \end{array} \tag{4.2.14}$$

By contrast, the computational effort for one iteration of the backpropagation algorithm consists of

$$\begin{array}{ll} \text{\# multiplications} & (2P+1)n_0n_1 + (5P+1)(n_1n_2 + \ldots + n_{l-1}n_l) \\ & +(P+1)(n_1+1) + (2P+1)(n_2 + \ldots + n_l) \\ \text{\# evaluations of } \sigma_L & P(n_1 + \ldots + n_l) \\ \text{\# evaluations of } \sigma_L' & P(n_1 + 2(n_2 + \ldots + n_l)) \end{array}$$

Thus, the computational effort for one iteration of the backpropagation algorithm as well as for one iteration of the conjugate gradient method is of magnitude $\mathcal{O}(P\,n)$ (see (4.2.2) for the definition of n).

4.3 Sensitivity Analysis

When the weights of a neural network are determined by the learning process then the knowledge of a neural network about the examined system is distributed among the different weight matrices and bias vectors. However, this distribution of the knowledge makes it difficult or even impossible to interpret the results generated by the neural network [19].

The representation of the activation value of a neuron in (4.1.3) shows that the outputs generated by a neural network depend on the input data $t_p \in I\!R^{n_0}$. In the following lemma, we show how the derivative of the output of the neural network with respect to the input data can be computed. Thus, it is possible to determine the sensitivities of the optimal output of the neural network with regard to small variations of the various components of the input vector t_p such that we can at least derive some information about the influence of each of the input data on the output of the neural network.

Lemma 4.3.1 *The sensitivities of the output of the neural network $y^l(w; t_p)$ with regard to the j-th component of the input data $t_p \in I\!R^{n_0}$ at a given point w are given by*

$$\sigma'(W^l y_p^{l-1} + \theta^l)\, W^l\, \sigma'(W^{l-1} y_p^{l-2} + \theta^{l-1})\, W^{l-1} \cdots \sigma'(W^1 t_p + \theta^1) W^1 e_j$$

with unit vectors $e_j \in I\!R^{n_0}$.

Proof: The derivative of the output of the neural network $y^l(w; t_p)$ with respect to the j-th component of the input t_p is given by

$$\frac{\partial y^l(w, t_p)}{\partial t_p} e_j .$$

Since $y^k(w, t_p) = y_p^k(w)$ is defined recursively by (4.1.3) and y_p^k depends only on y_p^{k-1} and $w^k = (W^k, \theta^k)$, an application of the chain rule yields

$$\frac{\partial y_p^l}{\partial t_p} = \frac{\partial y_p^l}{\partial y_p^{l-1}} \cdots \frac{\partial y_p^2}{\partial y_p^1} \cdot \frac{\partial y_p^1}{\partial t_p} . \tag{4.3.1}$$

Furthermore,

$$\frac{\partial y_p^k}{\partial y_p^{k-1}} = \sigma'(W^k y_p^{k-1} + \theta^k) W^k \in I\!R^{n_k \times n_{k-1}} , \qquad k = 2, \ldots, l , \tag{4.3.2}$$

and

$$\frac{\partial y_p^1}{\partial t_p} = \sigma'(W^1 t_p + \theta^1) W^1 \in I\!R^{n_1 \times n_0} . \tag{4.3.3}$$

Thus, inserting the expressions (4.3.2) and (4.3.3) into (4.3.1) yields

$$\frac{\partial y_p^l}{\partial t_p} e_j = \sigma'(W^l y_p^{l-1} + \theta^l) W^l \cdots \sigma'(W^1 t_p + \theta^1) W^1 e_j \in I\!R^{n_l} .$$

which completes the proof. ■

4.4 Numerical Results

The final section of this chapter deals with the application of neural networks in financial forecasting.

We start the section with verifying the convergence results presented in Section 3.2, see Subsection 4.4.1. Afterwards, we apply neural networks to forecasting the German stock index DAX on the basis of monthly data. In Subsection 4.4.2, we give a brief statement about the applicability of neural networks to financial forecasting. Then, in the following subsection, we explain the different aspects necessary to set up a neural network. We start the numerical experiments in Subsection 4.4.4 where we compare the Inexact Gauss–Newton Algorithm 2.2.3 with the traditional method for the training of neural networks, the backpropagation algorithm, which is a variant of steepest descent method. Subsequently, we discuss the quality of the forecasts generated by the neural network in Subsection 4.4.5. For the judgement of the forecasts, we use a graphical comparison of the forecasts of the neural network with the target value as well as the performance of a trading strategy which is based on the forecasts of the neural network. Then, in Subsection 4.4.6, we examine the sensitivities of the output data generated by the neural network in regard to small variations in the input data. Finally, at the end of this section, we give some concluding remarks on neural networks in financial forecasting.

4.4.1 Convergence Rates

To illustrate the convergence results of Section 3.2 we consider a 2–2–1 neural network, i.e. the input as well as the one hidden layer of the neural network contain two neurons and the output layer consists of one neuron. Using the notation introduced in Section 4.1, in this case the variable w has the following form

$$w = (W_{11}^1, W_{12}^1, W_{21}^1, W_{22}^1, \theta_1^1, \theta_2^1, W_{11}^2, W_{12}^2, \theta_1^2)\,.$$

The network is trained with 5 patterns of input and corresponding output data. Since the output of the network represents a prediction of a scaled value of the DAX, the input data of the network are scaled DAX values of the previous two months. A more precise description of the composition of the training patterns and the preparation of the input and output data will be given in the following subsection.

Iter. i	$\phi(w_i)$	$\|\nabla\phi(w_i)\|_2$	$\|w_i - w_*\|_i$	$\frac{\|w_i-w_*\|_i}{\|w_{i-1}-w_*\|_i}$	$\|w_i - w_*\|_2$	$\frac{\|w_i-w_*\|_2}{\|w_{i-1}-w_*\|_2}$
2	0.128D+00	0.694D−01	0.359D+01	0.61134	0.612D+01	0.79625
3	0.355D−01	0.119D+00	0.298D+01	0.62362	0.485D+01	0.79259
4	0.259D−02	0.394D−01	0.434D+01	0.95186	0.465D+01	0.95769
5	0.543D−03	0.210D−02	0.396D+01	0.95128	0.447D+01	0.96116
6	0.395D−03	0.449D−02	0.404D+01	0.96192	0.432D+01	0.96637
7	0.282D−03	0.924D−03	0.381D+01	0.90380	0.392D+01	0.90841
8	0.207D−03	0.207D−02	0.355D+01	0.91249	0.358D+01	0.91392
9	0.114D−03	0.103D−02	0.313D+01	0.89146	0.321D+01	0.89617
10	0.103D−03	0.323D−03	0.294D+01	0.96334	0.311D+01	0.96700
11	0.997D−04	0.747D−03	0.266D+01	0.89299	0.280D+01	0.90200
12	0.986D−04	0.211D−02	0.252D+01	0.93755	0.264D+01	0.94285
13	0.876D−04	0.677D−03	0.222D+01	0.87020	0.232D+01	0.87984
14	0.838D−04	0.219D−02	0.209D+01	0.92485	0.216D+01	0.92903
15	0.719D−04	0.741D−03	0.195D+01	0.91993	0.199D+01	0.92276
16	0.660D−04	0.591D−03	0.163D+01	0.82694	0.166D+01	0.83138
17	0.576D−04	0.249D−02	0.147D+01	0.89022	0.148D+01	0.89107
18	0.423D−04	0.506D−03	0.112D+01	0.75732	0.112D+01	0.75824
19	0.301D−04	0.179D−02	0.382D+00	0.34124	0.382D+00	0.34161
20	0.233D−04	0.328D−02	0.201D−01	0.05271	0.202D−01	0.05272
21	0.115D−05	0.664D−03	0.364D−04	0.00180	0.364D−04	0.00180
22	0.233D−10	0.358D−05	0.473D−09	0.00001	0.473D−09	0.00001

TABLE 4.1: Training of a 2–2–1 neural network with the inexact Gauss–Newton method presented in Section 2 ($\|x\|_i = \|J(w_i)^+ J(w_i)\, x\|$)

The results in Table 4.1 are generated by the Inexact Gauss–Newton Algorithm 2.2.3 presented in Chapter 2, i.e. the generated sequence has the form

$$w_{i+1} = w_i - J(w_i)^+ R(w_i) - J(w_i)^+ r_i\,. \tag{4.4.1}$$

Table 4.1 shows that this sequence converges to a solution w_* (see Column 6 of Table 4.1). The values of the last iterations in Table 4.1 make clear that the sequence (4.4.1) converges locally q–quadratically to w_* on $\mathcal{R}(J(w_i)^T)$. This corresponds to the theoretical results in Section 3.2 since the second column of Table 4.1 shows that the residual of the present problem at the solution is very small, probably even zero. Columns 4 and 6 of Table 4.1 illustrate that the difference of the current iterate w_i from the solution w_* in the null space of $J(w_i)$

$$\|(I - J(w_i)^+ J(w_i))(w_i - w_*)\|_2^2 = \|w_i - w_*\|_2^2 - \|J(w_i)^+ J(w_i)(w_i - w_*)\|_2^2$$

is very small. Therefore the q–quadratic convergence is obtained even on the whole space $I\!R^n$.

Iter. i	$\phi(w_i)$	$\|\nabla\phi(w_i)\|_2$	$\|w_i - w_*\|_i$	$\frac{\|w_i - w_*\|_i}{\|w_{i-1} - w_*\|_i}$	$\|w_i - w_*\|_2$	$\frac{\|w_i - w_*\|_2}{\|w_{i-1} - w_*\|_2}$
2	0.125D+00	0.778D−01	0.336D+01	0.60275	0.567D+01	0.77111
3	0.743D−01	0.124D+00	0.447D+01	0.79705	0.451D+01	0.79587
4	0.401D−01	0.147D+00	0.446D+01	0.99231	0.447D+01	0.99024
5	0.196D−01	0.316D−01	0.350D+01	0.78615	0.351D+01	0.78612
6	0.107D−02	0.191D−01	0.328D+01	0.93574	0.329D+01	0.93497
7	0.344D−03	0.395D−02	0.306D+01	0.93225	0.306D+01	0.93176
8	0.181D−03	0.258D−02	0.214D+01	0.70100	0.215D+01	0.70161
9	0.149D−03	0.605D−02	0.164D+01	0.76541	0.165D+01	0.76597
10	0.581D−04	0.156D−02	0.144D+01	0.87761	0.144D+01	0.87748
11	0.468D−04	0.167D−02	0.119D+01	0.82558	0.119D+01	0.82582
12	0.312D−04	0.926D−03	0.678D+00	0.57044	0.683D+00	0.57305
13	0.214D−04	0.238D−02	0.556D−01	0.08212	0.938D−01	0.13727
14	0.967D−05	0.166D−02	0.323D−01	0.57998	0.728D−01	0.77619
15	0.309D−07	0.188D−04	0.430D−04	0.00133	0.563D−01	0.77383
16	0.852D−11	0.664D−06	0.344D−04	0.88250	0.485D−01	0.86118
⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	0.335D−16	0.300D−08	0.155D−05	0.91021	0.891D−02	0.71941
25	0.235D−16	0.249D−08	0.763D−06	0.85563	0.571D−02	0.64054
26	0.167D−16	0.208D−08	0.265D−06	0.70877	0.275D−02	0.48114

TABLE 4.2: Training of a 2–2–1 neural network with the inexact Gauss–Newton sequence (4.4.2) ($\|x\|_i = \|J(w_i)^+ J(w_i)\, x\|_2$)

For a clear distinction between the convergence rate of the projected and unprojected iterates, consider the sequence

$$w_{i+1} = w_i - J(w_i)^+ R(w_i) - (I - J(w_i)^+ J(w_i)) z_i - J(w_i)^+ r_i \tag{4.4.2}$$

with $z_i = \left(\frac{1}{(i+1)^2}, \ldots, \frac{1}{(i+1)^2}\right)^T$. The convergence to a solution w_* is still obtained as well (see Table 4.2). However, in comparison with the convergence of the iterates in $I\!R^n$ (see columns 4 and 6 of Table 4.2), the convergence of the sequence (4.4.2) on $\mathcal{R}(J(w_i)^T)$ is much better.

Iter. i	$\phi(w_i)$	$\lVert\nabla\phi(w_i)\rVert_2$	$\lVert w_i - w_*\rVert_2$	$\frac{\lVert w_i-w_*\rVert_2}{\lVert w_{i-1}-w_*\rVert_2}$
2	0.796D−01	0.171D+00	0.275D+01	0.85624
3	0.261D−01	0.349D−01	0.199D+01	0.72504
4	0.113D−01	0.569D−01	0.171D+01	0.85794
5	0.150D−02	0.577D−02	0.289D+00	0.16861
6	0.383D−03	0.931D−02	0.175D+00	0.60685
7	0.250D−05	0.577D−03	0.751D−02	0.04287
8	0.158D−06	0.231D−03	0.671D−05	0.00089
9	0.793D−12	0.677D−06	0.798D−07	0.01189

TABLE 4.3: Training of a 2–2–1 neural network with the inexact Gauss–Newton method of Theorem 3.2.5

If the training of the described 2–2–1 network is carried out with the inexact Gauss–Newton method which generates the sequence of Theorem 3.2.5

$$w_{i+1} = w_i - J(w_i)^+R(w_i) - (I - J(w_i)^+J(w_i))(w_i - w_*) - J(w_i)^+r_i$$

we obtain the values presented in Table 4.3. Note that the problem is a zero–residual problem. Here, as in Theorem 3.2.5, the sequence w_i converges q–quadratically in norm to w_* without projecting it onto a proper subspace. Although this sequence of iterates cannot be used in general since the required information on w_* is not available, the numerical results underline the theoretical statements.

4.4.2 Neural Networks in Forecasting Stock and Index Prices

In this subsection, we summarize briefly some important aspects of financial forecasting and the application of neural networks in this area.

A model based examination of financial forecasting and the application of neural networks

In general, a forecast is a guess about one or more events that will happen in a future period and which is based on observations of the past [14, p. 551]. If forecasts are based on a model that gives a simplified description of the influence of observations of the past on the future value of the examined object, then the success of any forecast is determined by the quality and the temporal stability of this model [14], [53].

The quality of the model is determined by the extent and the faultless determination of the observations. These observations should include the examined object as well as all quantities which might have an influence on the future value of this object [14]. Note that the number of the observations causes the complexity of the model [5]. Furthermore, the quality of the model is based on the ability of a sufficiently good description of the influence of the observations on the future value of the examined object.

The temporal stability of a model is given if an explanation of the connection of the observations and the forecast which holds for a past period continues to be valid in the future [14]. However, the validity of this last aspect describes an universal problem since an exact repetition of past events does not exist [53]. Hence, the question is whether the structural change is so significant that the projection of valid structural connections from the past into the future would lead to false forecasts [53]. In addition to this Stöttner [100] mentions that a final solution of the dilemma of forecasts supported by observations of the past and the future structural instability is hardly possible and that this dilemma is finally the reason of an indelible remaining risk of any forecast.

Thus, the model for forecasting stock or index prices determines the quality of the forecast. Unfortunately, so far one failed to quote a mathematical description about the realization of a stock or index price such that it is not possible to assert a well defined model for the forecast of stock or index prices [5].

The stock market is marked by its highly complex structure whose co–operation of the variables is neither theoretically uniquely explainable nor empirically exactly describable [5]. Moreover, there exists a large variety of variables in the stock market which are not all obvious and which change over time. Furthermore, there is evidence that also the interactions of these variables change and that they are nonlinear [81].

Thus, for an efficient forecast of a stock or index price it is necessary to examine the linear and nonlinear dependencies of a large number of variables as well as their interactions.

The advantage of the application of neural networks in forecasting is that the underlying function form which generates the data does not need to be known in advance. The neural network rather tries to find this function during the learning process. In this spirit, the neural network learns the model for pricing the stocks and indices, respectively [65].

Complete freedom in the choice of the model, however, is also not given if neural networks are used. A certain idea of the model is necessary. In this connection, the choice of the data plays an important part: the data set should be sufficient large and representative concerning the task of stock or index price forecasting so that the structure of the formation of prices can be marked [5]. However, already the determination of a first idea of the model by choosing the variables, which might have an influence on the forecast, is difficult. Especially in financial forecasting, it is impossible to determine only roughly optimal a priori the number and the kind of preprocessing of the variables because of the complex interactions in the stock market [5].

A survey of methods for neural networks that deal with this phenomena is given in [5].

Specialization of the model underlying neural networks

In the following, we mark the models which are generated by neural networks.

As mentioned above, we restrict ourselves to the examination of feed–forward networks. These network architectures model the transition of the input into the output on the basis of the law of cause and effect since the information is transmitted through the network only in one direction. The transition between the input variables and the forecast is determined without an explicit influence of the time. Thus, the use of feed–forward architectures implies the formation of input/output models [5].

Neural networks also can be classified with regard to the different models that exist for the analysis of asset prices. The traditional explanation and valuation models for analyzing asset prices are divided regarding the origin of their variables and regarding the explanation of the formation of prices and the valuation approach for the calculation of prices. The existing two main model classes are the fundamental analysis and the technical analysis [16].

Fundamental analysis is the examination of the underlying forces that affect the well being of the economy, industry groups, and companies. To forecast future stock prices, fundamental analysis combines economic, industry, and company analysis to derive a stock's current fair value and forecast future value.

By contrast, technical analysis is the examination of past price movements to forecast future price movements. Technical analysts are sometimes referred to as chartists because they rely almost exclusively on charts for their analysis. Technical analysts believe that the current price fully reflects all information. Thus, the price represents the fair value and should form the basis for analysis.

Until now, the procedure was that the technical analysis and the fundamental analysis were both carried out independently. After their separated inquiry, the results are combined which is often not done methodically [5] but within the scope of qualitative discussions of experts.

Neural networks can be assigned either to the technical or to the fundamental approach by feeding exclusively either technical oriented or fundamental oriented input data into the neural network. The advantage of a technical or fundamental analysis with neural networks is that these allow the simultaneous analysis of very many variables at which additionally a nonlinear perspective can be established [5].

A further development of the traditional models for analyzing asset prices by neural networks consists in the combination of the fundamental and technical approach with reference to econometric models. In this case, the neural network simultaneously analyzes both variables which are assigned to the technical analysis and variables which are assigned to the fundamental analysis in solely one computation.

Hence, in neural networks the choice of the different approaches for analyzing stock or index prices consists in the determination of the input data, i.e. which variables and which preprocessing of the time series have to be chosen. According to the specification of the input data, a technical, fundamental or econometric model is supported.

4.4.3 Preparation of the Neural Network and the Training Patterns

Choice and preprocessing of the input data

The statements in the previous subsection illustrate that the selection of the data plays an important role in deriving an efficient model for forecasting stock or index prices. In general, the following aspects have to be considered for the choice of the data (see also [19]):

1. Do the data satisfy the requirements regarding quantity and quality?
2. Which data are suitable as input and output data with regard to the problem formulation and are added to the neural network?

3. In which form are the data to be presented to the neural network?

Regarding the quantity, it has to be guaranteed that a sufficient number of data sets are available, since the neural network gain their knowledge out of these data. The more extensive the data material is the better the different situation is covered and the smaller the runaways in the measured values is weighted [19]. However, one has to consider that the preprocessing of the data and the learning process gets more expensive with increasing data material. Furthermore, in this case the risk increases that contradictory information is added to the network which might make worse the learning result [19].

The quality of the data is the basis for the function built by the neural network, i.e. one has to clarify which are the possible values having a causal influence of the predicting value [87]. The choice of the data is certainly influenced by ones conviction regarding the success of technical or fundamental analysis in forecasting stock or index prices (see also the previous subsection). In this connection also the principle holds "garbage in – garbage out", i.e. the quality of the statements of the neural network largely depends on the validity and consistence of the data underlying the training process of the network [19]. However, as already mentioned above, often it is not possible to specify concretely which variables influence the examined object. Hence, it is necessary to include many variables into the model to increase the probability that the scope is described comprehensively.

The data which crucially determine the examined problem are to be chosen as the input data. The output of the neural network is determined by the wanted information [19]. In the case of stock or index price forecasting, the input vectors are the lagged values of a time series and the output is the prediction for the next value [39]

Moreover, within the scope of the representation of the data, we have to deal with the aspects of scaling and preprocessing of the data. The scaling of the data applies to the task that the data which are supplied to the network are mapped onto the interval which can be processed by the network [19]. The choice of the interval depends on the activation function. In our case, we have chosen the logistic function (see (4.1.1))

$$\sigma_L(r) = \frac{1}{1+e^{-kr}} \,.$$

For this function, it is necessary to scale the input and output data onto the interval $[0; 1]$. In doing so, we have chosen the interval $[0.1; 0.9]$ since the end points 0 and 1 are not reached by the logistic function. The preprocessing of the data deals with the question in which form the information should be made available to the neural network and which form the output of the neural network should have. In the field of financial forecasting, the preprocessing functions are often derived from technical analysis [78] in order to capture some of the underlying dynamics of the financial markets (see for instance [86], [109]). Regarding the output of the neural network, the preprocessing prescribes whether point predictions are carried out or the network output informs about the trend of the examined time series.

Construction of the training pattern

When the choice of the input and output data is made, the patterns for the training of the network have to be arranged. In this connection, we stress again how important the choice of the learning patterns is for the attainable efficiency of a neural network. The patterns in its entirety must represent as good as possible the set of all possible situations in the relevant part of the capital market. On the other side, the information in the training set must not get arbitrary large since the capability of learning, i.e. the storage capacity, in a neural network is limited.

Training:

	Input	Output
1. pattern	$(DAX_1\,, DAX_2\,, \$_1\,, \$_2)$	(DAX_3)
2. pattern	$(DAX_2\,, DAX_3\,, \$_2\,, \$_3)$	(DAX_4)
$\vdots$	$\vdots$	$\vdots$
10. pattern	$(DAX_{10}\,, DAX_{11}\,, \$_{10}\,, \$_{11})$	(DAX_{12})

Forecasting DAX_{13}**:**

	Input	Output
1. pattern	$(DAX_{11}\,, DAX_{12}\,, \$_{11}\,, \$_{12})$	**unknown**

FIGURE 4.5: Arrangement of the training and forecasting patterns

We will explain the arrangement of the training and forecasting patterns by the following example (see also Figure 4.5). Suppose we wish to forecast the German stock index DAX for instance for the month 1/1994, i.e. the closing value of the last day of the month 1/1994 (in Figure 4.5 this value corresponds to DAX_{13}). Furthermore, we assume that lagged values of the time series of the DAX itself and the US–Dollar are to present to the network, and that this are the values of the last twelve months each, i.e. the DAX and US–Dollar values of the months 1/1993 until 12/1993 (these are the values $DAX_1 - DAX_{12}$ and $\$_1 - \$_{12}$, respectively, in Figure 4.5). Each of the input vectors are to contain two scaled DAX values as well as two scaled US–Dollar values. Since each neuron in the input layer of the neural network only contains one value, thus, the first layer of the network consists of four neurons. The output of the pattern consists in a scaled DAX value according to the given problem formulation. According to the available data, we are now able to construct ten training patterns. The composition of this training patterns is made clear in Figure 4.5. This figure shows that the input vector contains the values of the DAX and US–Dollar which directly preceded the output value, i.e. if the output value for instance represents the DAX value of the month 5/1993 then the input vector of this pattern contains the DAX and US–Dollar values of the months 3/1993 and 4/1993.

After the termination of the training of the neural network, the actual vector of input data, i.e. the DAX and US–Dollar values of the months 11/1993 and 12/1993, is presented to the network in the example (this corresponds to the forecasting pattern in Figure 4.5). Then, the output of the neural network represents the prediction of the DAX value for the month 1/1994.

Note that the number of values in the input vector which belong to the same time series and the number of training patterns determine the period of observations that is available for the learning process.

The design of the neural network

The solution of the questions how many layers or how many units in any of the layers enables the best forecast is mathematically unsolved. It turned out that different results were achieved when the numbers of layers and units in any of the layer were varied; however, a deterministic method for the choice of the optimal network is not known so far. In principle, neural networks with only one layer of hidden neurons are already able to approximate any structure contained in a data set [78].

Regarding the number of neurons in each of the layer, we note that the number of neurons in the input and output layer are already determined by the decision which variables are to be presented to the network and the number of forecasts to be made with just one network, respectively. For the determination of the necessary number of units in the hidden layer, usually the strategy is followed such that one starts with a network whose dimension is too large for the given problem and subsequently removes superfluous units.

Thus, an optimal configuration of a neural network can only be found in an extensive trial–and–error process.

Overfitting

After the training patterns and the neural network architecture is fixed, the training process of the neural network, i.e. the adjustment of the weights in the neural network, is started.

Now, if the training process is carried out until the error of the outputs of the neural network and the target outputs in the training patterns is minimized, then the neural network strives to perfectly map the presented data. However, the extracted structure cannot be used for a generalization. Figure 4.6 makes clear that in case of a perfect map of the training patterns the neural network generally generates poor forecasts. This is the case because time series not only contain deterministic behavior but also a certain portion of noise that should not be modelled by the network. This problem of fitting the noise in addition to the signal is called overfitting [39].

Moreover, the problem of overfitting is increased by the constellation of a large number of parameters compared to a small number of training data. In problems like forecasting stock or index prices often there are only a few training data available compared to the complexity of the network parameter [109], especially when the adaption of the weights is carried out on the basis of monthly data [44]. This enables the network to obtain an arbitrary exact adaption to the training data without building an ability of generalization.

In forecasting, it is less important how well a model fits the training data. The aim of the training process rather is to achieve an optimal ability of generalization of the neural network, i.e. the network should extract regularities from the training examples that do transfer to new examples [19].

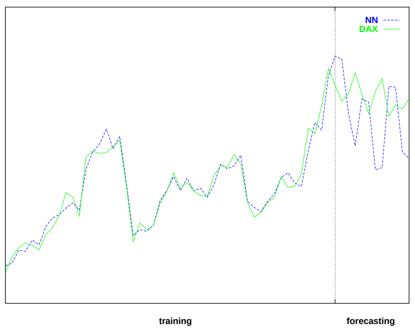


FIGURE 4.6: Training and Forecasting period of a neural network

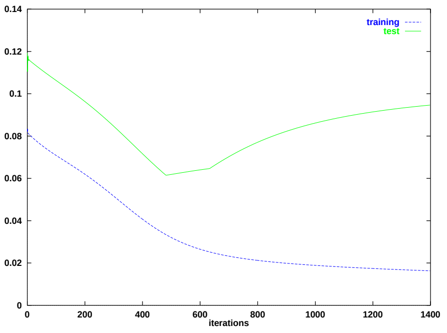


FIGURE 4.7: Error on the training set and the test set

Stopping criterion for the training process

One of the most well known techniques for attacking the overfitting problem [78] and achieving the ability of generalization of the neural network is the early stopping procedure, also called the Stopped–Training method.

In this method, the time when to stop during the learning process is determined when the ability of generalization of the network decreases. In this connection, the concept of cross validation has been proven successful. The idea of this concept is to divide the available patterns of input/output data into two disjoint sets (see e.g. Zimmermann [109]):

1. The set of training data serves for the actual learning process, i.e. the adaption of the unknown weights.
2. The set of test data is used after each iteration of the learning algorithm to compute the error generated by the network.

The plot of the error in predicting points out of the training set and the error in predicting points out of the test set in Figure 4.7 shows that the former error decreases monotonically during the complete training process, see the line in Figure 4.7 labelled with “training”. By contrast, the error on the test set will initially decrease as the network starts to learn the interactions of the input data, but then will begin to increase once the network starts to learn the noise, see the line in Figure 4.7 labelled with “test”. The location of the minimum of this error determines when the effective network complexity is right [39], i.e. this state of the neural network presents the best possible weights and accordingly should be saved and documented.

4.4.4 Training of Large–Scale Networks

For comparison of the inexact Gauss–Newton algorithm with the backpropagation algorithm, we carried out the training of different neural networks with these two methods. The networks considered have between 4,500 and 45,000 unknown parameters.

Network (# variab.)	Method	cpu in sec.	# of outer iter.	total # of cg iter.	aver. # of cg iter.
110–40–2 (4,522)	Backp.	5859	2031	-	-
	G.–N.	398	154	425	2.76
	G.–N. (mt)	296	66	343	5,20
110–80–2 (9,042)	Backp.	25161	4029	-	-
	G.–N.	952	239	383	1.60
	G.–N. (mt)	572	71	355	5.00
110–100–2 (11,302)	Backp.	82061	10000	-	-
	G.–N.	1899	398	598	1.50
	G.–N. (mt)	831	77	437	5.68
165–100–2 (16,802)	Backp.	24430	2743	-	-
	G.–N.	1304	249	374	1.50
	G.–N. (mt)	718	56	318	5.68
165–120–2 (20,162)	Backp.	108876	10000	-	-
	G.–N.	2612	429	604	1.41
	G.–N. (mt)	714	43	273	6.35
165–140–2 (23,522)	Backp.	74587	5853	-	-
	G.–N.	5225	730	1100	1.51
	G.–N. (mt)	1425	62	528	8.52
165–160–2 (26,882)	Backp.	124255	8104	-	-
	G.–N.	5608	701	1012	1.44
	G.–N. (mt)	1441	66	425	6.44
165–180–2 (30,242)	Backp.	168884	10000	-	-
	G.–N.	6080	709	866	1.22
	G.–N. (mt)	3244	200	703	3.52

TABLE 4.4: Comparison of the inexact Gauss–Newton method with the Batch-Backpropagation with Armijo line search (Backp. = Backpropagation, G.–N. = Inexact Gauss–Newton Algorithm 2.2.3 of Sec. 2, G.–N. (mt) = Inexact Gauss–Newton Algorithm 2.2.3 with a modified trust–region strategy)

The input data of these networks consist of the DAX and other time series such as e.g. the Dollar, the REX performance index, dividend yields. The goal of these neural networks is to predict the DAX value of the two following months. Therefore, the output layer of these networks contains two neurons. For the training of the network, 180 patterns are available.

As a termination criterion for the overall method, we used the Stopped–Training method described in the previous Subsection 4.4.3 to training the neural networks.

The results of the training of each of these networks with a backpropagation algorithm combined with a Armijo step size rule and the Inexact Gauss–Newton Algorithm 2.2.3 described in Chapter 2 are listed in the first two rows in Tables 4.4 and 4.5. These results show that the computation time of the inexact Gauss–Newton method in most of these examples is less than 5% of the computation time of the backpropagation.

Network (# variab.)	Method	cpu in sec.	# of outer iter.	total # of cg iter.	aver. # of cg iter.
220–100–2 (22,302)	Backp.	32782	3461	-	-
	G.–N.	3128	483	931	1.93
	G.–N. (mt)	1000	59	438	7.42
220–120–2 (26,762)	Backp.	117513	10000	-	-
	G.–N.	4849	690	1077	1.56
	G.–N. (mt)	1387	56	502	8.96
220–140–2 (31,222)	Backp.	40591	2961	-	-
	G.–N.	3846	472	696	1.47
	G.–N. (mt)	1121	81	264	3.26
220–160–2 (35,682)	Backp.	162602	10000	-	-
	G.–N.	7572	828	1235	1.49
	G.–N. (mt)	1728	62	485	7.82
220–180–2 (40,142)	Backp.	69421	3831	-	-
	G.–N.	7577	775	986	1.27
	G.–N. (mt)	2594	93	625	6.72
220–200–2 (44,602)	Backp.	167536	8135	-	-
	G.–N.	9244	852	1063	1.25
	G.–N. (mt)	1702	64	347	5.42

TABLE 4.5: Comparison of the inexact Gauss–Newton method with the Batch-Backpropagation with Armijo line search (Backp. = Backpropagation, G.–N. = Inexact Gauss–Newton Algorithm 2.2.3 of Sec. 2, G.–N. (mt) = Inexact Gauss–Newton Algorithm 2.2.3 with a modified trust–region strategy)

The results listed in the last row of each subsection in Tables 4.4 and 4.5 are obtained with the inexact Gauss–Newton method but with a slightly modified trust–region strategy. Usually, the trust–region radius is updated with the rule

$$\Delta_{i+1} = \begin{cases} \frac{\|\delta w_i\|}{2} & , \text{ if } \rho_i \leq \alpha_1, \\ \min\{2\,\|\delta w_i\|\,, \bar{\Delta}\} & , \text{ otherwise.} \end{cases}$$

In the modified version the increase of the trust–region is more aggressive. For $\rho_i > \alpha_1$ the radius is not dependent anymore on the current direction but on the radius of the previous trust–region and accordingly we set

$$\Delta_{i+1} = \begin{cases} \frac{\|\delta w_i\|}{2} & , \text{ if } \rho_i \leq \alpha_1, \\ \min\{2\,\Delta_i\,, \bar{\Delta}\} & , \text{ otherwise.} \end{cases}$$

Additionally, in the modified version of the trust–region strategy the stopping criterion when reaching the trust–region is changed slightly. If a conjugate gradient direction reaches the boundary of the trust–region, then the ratio ρ_i of the actual decrease to the predicted decrease is computed. If $\rho_i \leq \alpha_2$ we proceed as described in Algorithm 2.2.3 in Chapter 2. Otherwise, we ignore the current trust–region and continue until the conjugate gradient iterate fulfills $\rho_i \leq \alpha_2$. Hence, this modified version considers not only the previous quadratic model for the adjustment of the trust–region but also the current one.

The numerical tests show that this strategy accelerates the inexact Gauss–Newton method even further, as the number of outer Gauss–Newton iterations is reduced. A consequence of the modified trust–region strategy is that the average number of conjugate gradient iterations increases but due to the large reduction of the number of Gauss–Newton iterations the total number of conjugate gradient iterations is also decreased.

Network (# variab.)	# of train. pattern	cpu in sec.	# of outer iter.	total # of cg iter.	aver. # of cg iter.	approx. cpu of 1 cg iter.
390–195–1 (76,441)	900	49046	97	1645	16.96	29.82
	1200	81052	114	2207	19.36	36.72
	1800	208766	152	3932	25.87	53.09
520–260–1 (135,721)	900	114659	105	2656	25.30	43.17
	1200	144506	110	2542	23.11	56.85
	1800	485480	151	5938	39.32	81.76
845–425–1 (359,976)	900	264923	84	2794	33.26	94.82
	1200	316647	113	2384	21.10	132.82
	1800	956468	137	4541	33.15	210.63

TABLE 4.6: Training of very large neural networks with the inexact Gauss–Newton method using the modified trust–region strategy

Finally, Table 4.6 contains results of the training of very large neural networks (i.e. up to 360,000 unknowns) using the Inexact Gauss–Newton Algorithm 2.2.3 with the modified trust–region strategy. The input of these networks contains data of different time series which are based on daily data. The reason for the large increase of the computation time of the training of these large networks compared to that of the networks of middle size in Tables 4.4 and 4.5 is, apart from the increase of the number of variables, the larger number of training patterns. The latter causes the increase of the average number of conjugate gradient iterations and the increase of the approximate computation time of one conjugate gradient iteration (cpu time / total # of conjugate gradient iteration) which almost depends linearly on the number of training patterns assuming a fixed network (see also the theoretical statements about the computational effort in Section 4.2).

Furthermore, in Figures 4.8 and 4.9 we have plotted the computation time of the training for neural networks of middle and large size in dependence of the number of unknown weights in the neural network. These plots show that the computation of the training also depends almost linear on the number of weights in the neural network. Moreover, the Figures 4.8 and 4.9 show that the increase of the computation time is at a smaller rate for the inexact Gauss–Newton method with the modified trust–region strategy.

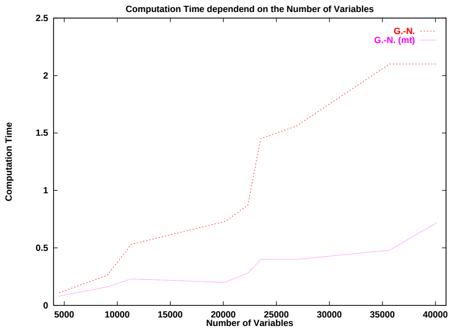


FIGURE 4.8: Computation time of networks of middle size

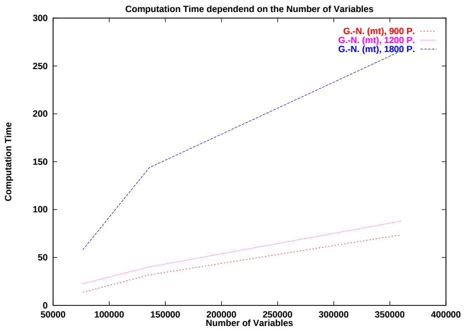


FIGURE 4.9: Computation time of networks of large size

The results in this subsection about the training of neural networks using the Inexact Gauss–Newton Algorithm 2.2.3 show that this method enables also a convenient use of middle and large scale neural networks.

4.4.5 Forecasting and Performance

The goal of the development of neural networks is to find a network that achieves results that are as good as possible on unknown patterns, i.e. with data which was not available during the training process. Therefore, it is necessary to compare and review the results of different developed networks on the basis of so–called forecasting data after the training process is finished. In this connection, input/output patterns are put aside that are not used in the training process and that serve to test the quality of the forecasts of each of the neural networks. These forecasts that are made for a past period to test the quality of the underlying model are so–called ex post forecasts [14]. To illustrate the procedure of testing the quality of neural networks, suppose that for instance in January 1995 a forecasting model is developed to predict the future value of the DAX on monthly basis. Assume that therefore the underlying system of the process describing the formations of prices is examined by observations belonging to the period starting January 1992 until December 1993. Then, for the period from January 1994 until January 1995, the future values are forecasted although there already exists observations for this period. These ex post forecasts serve for the check of the model. Subsequently, if the forecasting model proves to be good, it is used to forecast ex ante the still unknown values for February 1995 and the following months (see [14]).

Whereas during the training period, usually the mean squared error is used for the measurement of the learning progress, we use different reference numbers for the ex post judgement. A central difficulty in the judgement of the quality of a forecasting model is the choice of the proper values, that form the basis of the valuation, as well as a proper benchmark [87]. In the following we compare the output computed by the network with the target output. Furthermore, we use the rate of right trends and the trading profit for a valuation of each of the model.

Although we perform point prediction, we still can deduce information from the output of the neural network about the trend of the price process of the examined asset. If the future value predicted by the neural network is larger than the newest value of the corresponding time series in the input layer, then the network predicts an increase and in the opposite case the prediction of the network is interpreted as a decrease of the price of the examined asset. Then, for the computation of the rate of right trends one only needs to count the number of right and false trends given by the neural network to compute the rate of the right forecasts.

The computation of the trading profit is more complicated. For this a trading model (which converts predictions to recommendations for actions [39]) has to be incorporated into the simulation. In our simulations, we will use the following simple trading strategy: If the neural network predicts an increasing price, then the trading model orders to buy fictitiously the index (or to hold the index, if already a long position exists). In the opposite case, i.e. the network predicts a decreasing price, then the trading model orders to sell the index and to invest in bonds or time deposits (or to do nothing if already an investment in bonds or time deposits exists). The profit from this trading strategy under consideration of transaction costs is then compared with the profit of a simple buy–and–hold strategy. Note that the latter strategy also reflects the general market development in the examined time period (see also [87]); and, hence, is appropriate to serve as a reference value for a assessment of the trading profit of the neural network.

In our simulation, we leave the network architecture fixed for the simulation period. However, for each forecast we start the training process again to renew the weights in the neural network so that also the more recent dynamics in the corresponding market is incorporated.

net	input	sign acc.	perf.
10-10-1	dax	67 %	167
30-6-1	dax, us, rex	67 %	187
40-5-1	dax, us, rex, div	72 %	196
50-5-1	dax, us, rex, div, cuy	80 %	215

TABLE 4.7: Dependence of the choice of the input data on the quality of the performance of the neural network (dax = DAX, us = US–Dollar, rex = REX performance index, div = dividend yields, cuy = current yield)

Table 4.7 lists the results about the rate of right trends and the performance of the above described trading strategy for different neural networks. The second column informs about the different input data of the neural network. The numbers in the third column are the rate of right trends attained by each of the neural networks. Furthermore, the last column of Table 4.7 informs about the performance of the trading strategy of each of the neural networks for an initial investment amount of 100 DM. The simulation period is January 1992 until December 1994. The task of each of the networks is the prediction of the future DAX price. One can see that both numbers increase if the network is not only fed with DAX prices but also with other information which could have an influence on the development of the DAX price. These results underline the statements in Subsection

4.4.3 and make clear that a naive application of neural networks will fail for financial data (see also [39]).

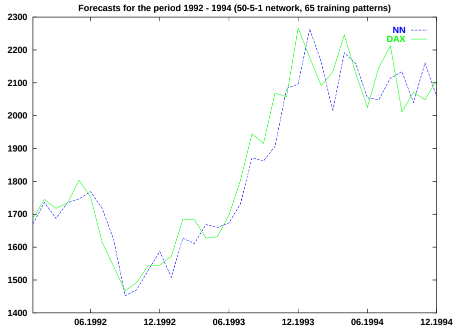


FIGURE 4.10: Forecasting of the neural network

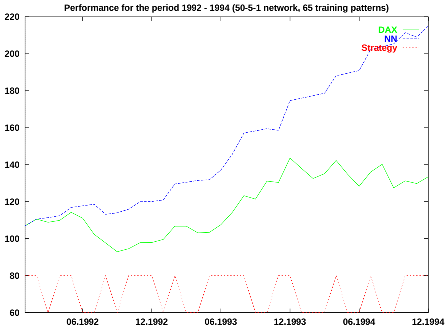


FIGURE 4.11: Comparison of NN–portfolio and DAX–portfolio

In Figure 4.10 we have plotted the forecast of the last neural network listed in Table 4.7 against the target output (i.e. the particular DAX value) for the period January 1992 until December 1994. Figure 4.10 shows that some of the predictions of the network are very close to the particular DAX value (see f.e. the prediction for January 1992, February 1992 and June 1992) however in other months the difference between the output of the neural network and the target output is quite large (this is especially the case at the end of the simulation period).

Although it would be most profitable to know the exact future price of the examined asset, a rough estimation is often enough to make a reasonable profit on dealing operations. In Figure 4.11, we have plotted the performance of the trading strategy underlying the above mentioned 50–5–1 neural network (i.e. the line labelled with “NN”) against the performance of a buy–and–hold strategy (i.e. the line labelled with “DAX”). Although this neural network has failed to accurately predict the particular DAX value, the performance of the neural network trading strategy beats the benchmark (i.e. the general market development) quite clearly in the period January 1992 until December 1994. A prerequisite for a good performance is that the neural network detects decreases of the price to avoid losses in the investment. Figure 4.11 makes clear that the examined network has identified all except three of these situations. Furthermore, for a good performance like the one in Figure 4.11 additionally it is necessary to participate in price increases of the examined asset. The lower line in Figure 4.11 that jumps between 60 and 80 informs about the current type of the investment. The line jumps to or stays at 60 , respectively, if the trading strategy gives the order to invest in bonds or time deposits. If the line jumps to or stays at 80 then the order is to invest in the index. This line, that is labelled with “Strategy”, shows that the examined 50–5–1 network in general identifies also price increases so that in all the trading strategy underlying this neural network achieves this excellent performance.

This illustrates that for speculative purposes, only trend turning points are actually important.

Therefore, many of the financial institutions which are trying to predict future asset prices generally rate algorithms by evaluating the percentage of time when the algorithm gives the right trend from today until some time in the future [68].

net	input	# of train.–pattern	# in Fig. 4.13 & 4.15
10–10–1	dax	65	44
20–8–1	dax, us	65	48
30–6–1	dax, us, rex	65	52
40–5–1	dax, us, rex, div	65	56
50–5–1	dax, us, rex, div, cuy	65	60

TABLE 4.8: List of neural networks used for backtesting (dax = DAX, us = US–Dollar, rex = REX performance index, div = dividend yields, cuy = current yield)

However, we also noticed that the performance of this 50–5–1 network was worse for other periods. Since the conditions in financial markets change dynamically it is necessary that the model approximating the particular market has to be adapted correspondingly. To possibly react more flexible on these structural changes in the market we carried out a so–called backtesting. In this connection, we examined a set of neural networks with different network architectures as well as a different choice of input data by measuring the performance of the trading strategy underlying each of these networks for an in advance defined period of time. For example for a prediction of the DAX of January 1994, we measured the performance of the trading strategy of, say, five different networks for the period starting January 1993 until December 1993. Then, the neural network which has performed best on these past data is used for the forecast of the next future DAX price, i.e. the DAX at the end of January 1994 in our example. For the following forecast, i.e. the forecast of the DAX in February 1994, the whole process is repeated all over, however, with all the dates put off by one month into the future.

In the following, we show the results of backtesting simulations for the period January 1990 until December 1994. In these simulations the set of neural networks consists of the networks listed in Table 4.8.

In the first simulation, the performance of each of the different networks was measured for a period of twelve months. Figure 4.12 shows the forecasts and Figure 4.13 displays the performance of an investment following the trading strategy of this backtesting simulation. The choice of the different networks for each of the forecasts is given by the lowest line in Figure 4.13. The relation of the numbers and the different networks is given in the last column of Table 4.8. The quality of the forecasts and the performance of the networks chosen in this backtesting simulation is certainly not as good as the forecasts and the performance of the 50–5–1 network in Figures 4.10 and 4.11 (note that the period of the backtesting simulation is longer), but in this backtesting simulation most of the time the trend turning points were found in time, too, so that the losses in the investment only

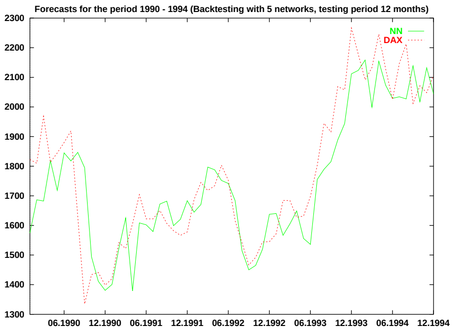


FIGURE 4.12: Forecasting of the neural network

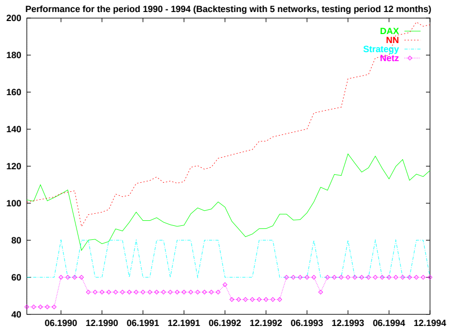


FIGURE 4.13: Comparison of NN–portfolio and DAX–portfolio

occurred for a few times and only to a small amount. The performance of the networks chosen in the backtesting simulation is not that outstanding than the performance of the 50–5–1 network since in the backtesting simulation the underlying investment has not always participated in price increase.

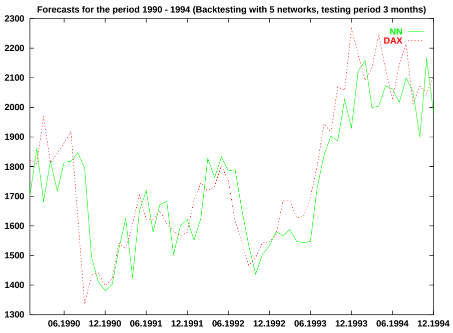


FIGURE 4.14: Forecasting of the neural network

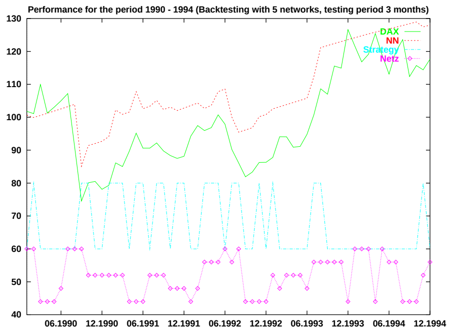


FIGURE 4.15: Comparison of NN–portfolio and DAX–portfolio

Unfortunately, these good results also are not generally obtained. If the performance measurement is carried out for a period of three months, the performance of the networks based on the backtesting is much worse (see Figures 4.14 and 4.15). At the end of the examined period the trading strategy based on the networks chosen in the backtesting simulation is only slightly better than the buy–and–hold strategy.

4.4.6 Sensitivities

Economists are often not only interested in the forecast by itself but also in the relation inherent in the financial markets. The knowledge found during the training of the neural network is distributed among the different weights and biases of the neural network. In general, it is very difficult to extract and interpret this knowledge. One attempt to make statements about the found knowledge is the analysis of the sensitivities to identify relations between the input data and the forecast. In Section 4.3, we have shown that the sensitivities of the output of the neural network $y^l(w;t_p)$ with regard to the jth component of the input data $t_p \in I\!R^{n_0}$ at a given point w are given by

$$\frac{\partial}{\partial (t_p)_j} y^l(w;t_p) = \sigma'(W^l y_p^{l-1} + \theta^l)\; W^l\; \sigma'(W^{l-1} y_p^{l-2} + \theta^{l-1})\; W^{l-1} \cdots \sigma'(W^1 t_p + \theta^1) W^1 e_j$$

with unit vectors $e_j \in I\!R^{n_0}$ and $(t_p)_j$ being the jth component of the input data t_p.

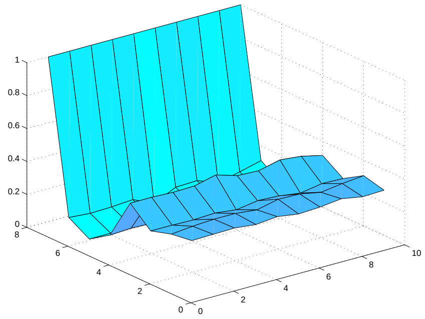


FIGURE 4.16: Sensitivities with respect to the time of the input data of a 8–8–1 network; the training was carried out only once

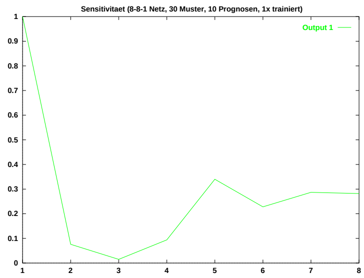


FIGURE 4.17: Average sensitivities with respect to the time of the input data of a 8–8–1 network; the training was carried out only once

In Figure 4.16, we have plotted the sensitivities of a 8–8–1 network which only contained DAX values in the input layer. The training of the network was only carried out once such that the forecasts of the next ten dates are all made on the basis on the same weights and biases of the network but on different input data. This aspect is reflected by the plot of the sensitivities as the sensitivities for the different input pattern are almost constant. One can also see that the latest DAX value, which corresponds to 8 on the x–axis, has the largest influence on the forecast, i.e. small changes in the latest DAX value will cause the largest changes in the output of the neural network. Figure 4.17 contains the average sensitivities over the different forecasting patterns which are given by

$$\frac{1}{\tilde{P}} \sum_{p=1}^{\tilde{P}} \frac{1}{s^p_{\max}} \frac{\partial}{\partial t_p} y^l(w;t_p)$$

with $\tilde{P}$ is the number of forecasts and

$$s^p_{\max} = \max_{j=1\ldots,n_0} \left| \frac{\partial}{\partial (t_p)_j} y^l(w;t_p) \right| , \qquad p = 1\ldots,\tilde{P}\,.$$

Figures 4.18 and 4.19 display the sensitivities of the same 8–8–1 network as above but in contrast to Figures 4.16 and 4.17 the training was carried out and, thus, the weights and biases were

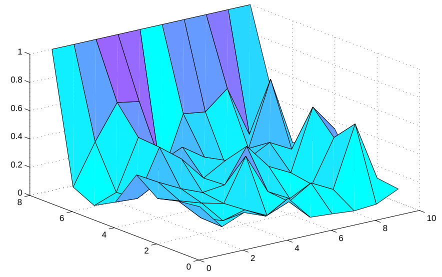


FIGURE 4.18: Sensitivities with respect to the time of the input data of a 8–8–1 network; the training was carried out for each forecast

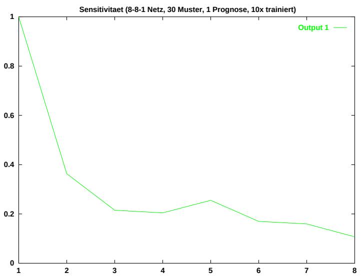


FIGURE 4.19: Average sensitivities with respect to the time of the input data of a 8–8–1 network; the training was carried out for each forecast

adapted, for each of the ten forecasts. Figure 4.18 makes clear that again the latest DAX value has the largest influence on each of the forecasts, but the size of the influences of the other input data changes during the simulation period. Furthermore, Figure 4.17 shows that older input data have less influence on the forecast.

In Figures 4.20 and 4.21 we plotted the sensitivities with respect to the time of the input data of the 50–5–1 network which we have already examined in the previous subsection. If the composition of the different input data in the input vector t_p is given by

$$t_p \;=\; (t_p^1\,, t_p^2\,, t_p^3\,, t_p^4\,, t_p^5)^T$$

with t_p^i containing the different time series values of the ith type of input data, i.e.

$$t_p^i = ((t_p^i)_1\,, (t_p^i)_2\,, \dots, (t_p^i)_{10})^T\,, \quad i = 1\,, \dots, 5\,,$$

then the sensitivities with respect to time given in Figures 4.20 are computed according the formula

$$\tfrac{1}{5}\sum_{j=1}^{5} \frac{\partial}{\partial t_p^j} y^l(w;t_p)\,, \quad p = 1\,\dots, \tilde{P}\,,$$

and the ones in Figure 4.21 are given by

$$\tfrac{1}{\tilde{P}}\sum_{p=1}^{\tilde{P}} \tfrac{1}{\tilde{s}_{\max}^p} \sum_{j=1}^{5} \frac{\partial}{\partial t_p^j} y^l(w;t_p)$$

with $\tilde{P}$ is the number of forecasts and

$$\tilde{s}_{\max}^p = \max_{i=1\,\dots,10} \left| \sum_{j=1}^{5} \frac{\partial}{\partial (t_p^j)_i} y^l(w;t_p) \right|\,, \qquad p = 1\,\dots, \tilde{P}\,.$$

Figure 4.21 shows that the newest input data again have the largest influence on the forecast of the neural network, however, the difference to the older input data is not as large as in the previous Figures 4.16 – 4.19. It is possible that regarding the other time series (i.e. DAX, REX Performance

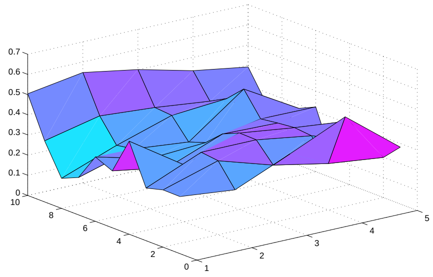


FIGURE 4.20: Sensitivities with respect to the time of the input data of a 50–5–1 network with different type of input data

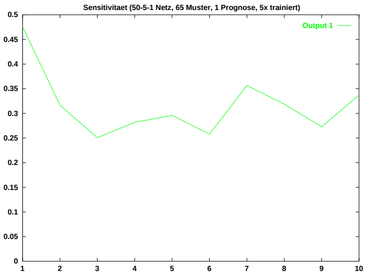


FIGURE 4.21: Average sensitivities with respect to the time of the input data of a 50–5–1 network with different type of input data

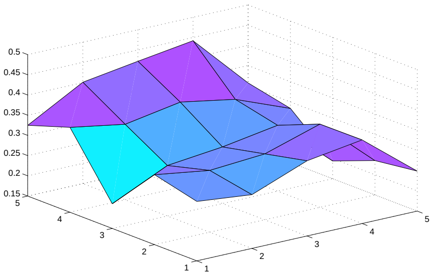


FIGURE 4.22: Sensitivities with respect to the type of the input data of a 50–5–1 network

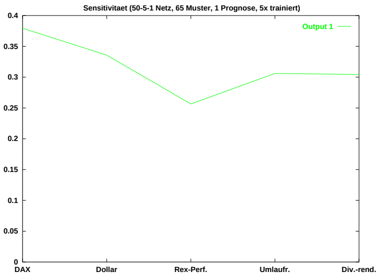


FIGURE 4.23: Average sensitivities with respect to the type of the input data of a 50–5–1 network

etc.) the difference of the influence with regard to the time of the input data is not as large as for the DAX time series.

Finally, in Figures 4.22 and 4.23, the sensitivities of the 50–5–1 network with respect to the different input data are shown. The sensitivities displayed in Figure 4.22 are computed according to the formula

$$\frac{1}{10}\sum_{i=1}^{10}\frac{\partial}{\partial(t_p)_i}y^l(w;t_p)\,,\qquad p=1\,\ldots,\tilde{P}\,,$$

with $(t_p)_i = ((t_p^1)_i\,,(t_p^2)_i\,,(t_p^3)_i\,,(t_p^4)_i\,,(t_p^5)_i)$ containing the ith time series value of each of the different type of input data. Furthermore, the average sensitivities in Figure 4.23 are given by

$$\frac{1}{\tilde{P}}\sum_{p=1}^{\tilde{P}}\frac{1}{\hat{s}^p_{\max}}\sum_{i=1}^{10}\frac{\partial}{\partial(t_p)_i}y^l(w;t_p)$$

with $\tilde{P}$ is the number of forecasts and

$$\hat{s}^p_{\max} = \max_{j=1\,\ldots,5}\left|\sum_{i=1}^{10}\frac{\partial}{\partial(t_p^j)_i}y^l(w;t_p)\right|\,,\qquad p=1\,\ldots,\tilde{P}\,.$$

This plot reflects the expected result that the DAX value has the largest influence on the development of the time series of the DAX.

4.4.7 Concluding Remarks on Neural Networks in Financial Forecasting

A critical item in using neural networks in forecasting is to find an optimal neural network for the given task. This process is very time–consuming since not only the choice of the input data but also the decisions about the number of hidden layers of the network as well as the number of neurons in each of these layers have to be made. Furthermore, if an "optimal" network is found that has obtained good results for a specific period it necessarily does not work well in other periods since the capital markets are subject to permanent structural changes of the interactions of the different variables. Hence, the set of input data and the network architecture must permanently be adapted to consider the changes in the capital markets. Nevertheless, neural networks have generated some very promising results in forecasting financial prices, see for instance the results in Subsection 4.4.5.

Unfortunately, at the beginning of the application of neural networks in financial forecasting, some expected that neural networks would almost be equivalent to the legendary "crystal ball". However, this expectation could not be fulfilled by neural networks, since they also only generate their knowledge on the basis of past data. In the end, however, neural networks are a promising attempt when unknown linear and nonlinear connections in the examined system have to be treated and when data in sufficient quantity and quality are available.

Chapter 5

Constructing the Instantaneous Volatility Function

In this chapter, we are concerned with the valuation of European call options. We will show in the first section that an important parameter for determining the value of the European call option is the volatility parameter σ. For a long time, this parameter was assumed to be a constant according to the well known model of Black and Scholes. However, it is evident that the assumption of a constant volatility parameter in the model describing the underlying asset price process is not conform to reality since the computation of implied volatilities from market European call values has shown that the volatility varies with the strike price as well as with the maturity of the option. In this work, we will examine a European option model that is consistent with this volatility smile. In this model the volatility parameter now represents a function dependent on the strike price and the maturity of the European call option. In Section 5.2, we will present this model that represents a second–order parabolic initial–value problem and show that the value of the European call option is the unique solution to this problem. Then, in the following section, we will introduce the mathematical formulation of the construction of the volatility function $\sigma(K,T)$ dependent on the strike price K and the maturity T of the option. The resulting optimization problem represents an inverse problem with the system equation given by a parabolic initial–value problem.

Treating this problem numerically requires a discretization of the parabolic initial–value problem which will be presented in the following Section 5.4 by using a finite difference method. Moreover, we examine the convergence behavior of this finite difference method and introduce the finite dimensional version of the optimization problem describing the construction of the volatility function. This problem represents an underdetermined nonlinear least–squares problem. Applying the Inexact Gauss–Newton Algorithm 2.2.3 to this problems requires the evaluation of the Jacobian of the residual function of the nonlinear least–squares problem times a vector as well as the evaluation of its transpose times a vector. In Section 5.5, we will present subroutines for both of this evaluations without storing the Jacobian explicitly. We end this chapter by presenting some numerical results for a synthetic European call option example and for constructing the volatility function for the German stock index DAX.

5.1 Motivation

European call and put options are a special kind of derivative securities. These are securities whose value depends on the values of other more basic underlying variables [58] and which can establish rights as well as obligations [92]. Strictly speaking, a European call option gives the holder the right to buy an underlying security (the underlying) for a fixed price K (the strike price or exercise price) on a fixed date T (the maturity, expiration date or exercise date), see for instance Hull [58] or Wilmott, Dewynne, and Howison [107]. A European put option, on the other side, gives the holder the right to sell an underlying security for a fixed price K on a fixed date T [58], [107]. The holder of a European call or put option only will exercise the option if the strike price K of the option is smaller or larger, respectively, than the value of the underlying asset S at the expiration date T. Since options only establish rights and no obligations for the holder these derivative securities have a certain price. The value of the European call and put option at time t will be denoted by $C(S,t;K,T)$ and $P(S,t;K,T)$, respectively. It is well known that the value of European call and put options are closely related to each other according to the put–call parity

$$S + P(S,t;K,T) - C(S,t;K,T) = K\exp(-r(T-t)),$$

if equal borrowing and lending rates are assumed and with the risk–free interest rate denoted by r, see for instance Merton [73]. Therefore, in the following we will only examine the valuation of European call options.

For the valuation of options there exist different models. In general, they establish a relationship between the traded derivative, the underlying asset and some market variables, e.g. volatility of the underlying asset [9], [73].

A well known model in this context, that is often used in financial practice, is the Black–Scholes model [9]. This model assumes that the value S of the asset underlying the option satisfies the following stochastic differential equation (a geometric Brownian motion with constant process parameters)

$$dS = \mu\, S\, dt + \sigma\, S\, dW_t \tag{5.1.1}$$

where the constants μ and σ are the drift and the volatility parameter of the asset price process, respectively. The random part of the price process is described by a Brownian motion W_t. Under this assumption, the price of a European call option $C(S,t;K,T)$ can be described by the following deterministic differential equation

$$C_t + \tfrac{1}{2}\,\sigma^2\, S^2\, C_{SS} + (r-d)\, S\, C_S = r\, C \tag{5.1.2}$$

with the end condition

$$C(S,T;K,T) \;=\; \max(S-K,0)\,. \tag{5.1.3}$$

The parameter d in (5.1.2) represents the constant continuous dividend yield of the underlying asset. Since the parabolic differential equation (5.1.2) can be transformed to the one–dimensional heat equation through simple transformation of the variables there exists even an explicit representation of the value $C(S,t;K,T)$ of a European call option in the Black–Scholes model. This so–called Black–Scholes formula is given by

$$\begin{aligned} C(S,t;K,T) &= S\ \exp(-d(T-t))\ \Phi(a_1(S,t;K,T)) \\ &\qquad -K\ \exp(-r(T-t))\ \Phi(a_2(S,t;K,T)) \end{aligned} \tag{5.1.4}$$

with

$$\begin{aligned} a_1(S,t;K,T) &= \frac{\ln(\frac{S}{K}) + (r-d+\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \\ a_2(S,t;K,T) &= a_1(S,t;K,T) - \sigma\sqrt{T-t} \end{aligned}$$

and where

$$\Phi(x) = \int\limits_{-\infty}^{x} \tfrac{1}{\sqrt{2\pi}}\ \exp(-\tfrac{u^2}{2})\ du$$

is the standard normal distribution function. The parameters in the Black–Scholes formula (5.1.4) are all but the volatility parameter σ, in principle, directly observable in the financial market. An estimation of the volatility parameter σ can be done in different ways. One possibility is to estimated the volatility empirically from historical data. A more common way is to compute the volatility parameter σ implied by an option price observed in the market by numerically inverting the Black–Scholes formula (5.1.4), see e.g. Hull [58].

Despite the widespread use of the Black–Scholes model in financial practice, it is generally realized that the assumptions of the model are not all conforming to reality. The existence of term structures in interest rates and dividends, for instance, indicates that r and d are not constants but (at least) functions of t and T [1]. Assuming a constant volatility σ, the Black–Scholes formula has in this case the form (see for instance Andersen and Brotherton–Ratcliffe [1])

$$\begin{aligned} C(S,t;K,T) &= S\ \exp\left(-\int_t^T d(\tau)d\tau\right)\ \Phi(a_1(S,t;K,T)) \\ &\qquad -K\ \exp\left(-\int_t^T r(\tau)d\tau\right)\ \Phi(a_2(S,t;K,T)) \end{aligned} \tag{5.1.5}$$

with

$$\begin{aligned} a_1(S,t;K,T) &= \frac{\ln(\frac{S}{K}) + (\ \int_t^T (r(\tau)-d(\tau))d\tau\ + \frac{1}{2}\sigma^2\ )(T-t)}{\sigma\sqrt{T-t}} \\ a_2(S,t;K,T) &= a_1(S,t;K,T) - \sigma\sqrt{T-t} \end{aligned}$$

Moreover, Rubinstein observed in [88] that the Black–Scholes formula became increasingly unreliable over time. His examinations of S&P 500 index options on the Chicago Board Options Exchange (he expected that this market approximates best the assumptions of the Black–Scholes formula) showed that in 1976 to 1978 differences between the market price of these options and the value computed according to the Black–Scholes formula existed but were not economically significant. He examined the same for 1986. However, starting in 1987, the errors between market prices and option values computed with the Black–Scholes model increased rapidly, see Rubinstein

[88, Section 1]. It seems that the crash of October 1987 increased the likelihood assigned by market participants to extreme stock market movements [1]. Rubinstein [88] observed that put options with strike prices smaller than the current asset price, i.e. out–of–the–money puts, were priced much more highly. Since the Black–Scholes formula is monotonically increasing in volatility this means that options with a low strike price had significantly higher implied volatilities than high striking price options. Furthermore, the examination of options with different maturities has shown that the volatility parameter σ is also not constant with respect to the maturity, see for instance Dupire [32]. This phenomenon of a time- and strike–dependent volatility parameter σ is called the volatility smile or the volatility skew and indicates that the stock prices do not follow the log–normal distribution with constant drift and volatility parameter that is assumed in the Black–Scholes model [1].

In financial practice, these deficiencies were considered by managing tables of values for r, d and σ which are used with different option maturities and strikes. This approach works for European options with short maturities and strike prices which are close to the current asset price; but it is unsuitable for options which are far out or in the money and/or with long maturities. These options are not or only rarely traded such that there do not exist volatility values for them. However, financial institutes might also be interested in information about these volatilities since their portfolios can contain *older* positions that were held speculative so far and now, they wish to hedge these positions with options with strike prices that are not close to the current asset price. Furthermore, financial institutions have to deal with individual inquiries from customers. For the pricing of these individual products information might be necessary about volatilities of the market price of the underlying asset with respect to a long maturity. Moreover, the above approach does not work for the pricing of more complicated options such as exotic options or options with early exercise features since in these cases it is not clear which values for r, d, and σ should be used, see Andersen and Brotherton–Ratcliffe [1].

Many approaches exist which try to be consistent with the existence of a volatility smile. Merton [74], for instance, expands the Black Scholes model by introducing Poisson jumps into the model. Hull and White [59] are examining options on assets that have a stochastic volatility. Unfortunately, these approaches introduce an additional source of risk so that they are not complete anymore. Thus, the concept of arbitrage–free pricing cannot be applied and assumptions about investor preferences and behavior have to be made.

However, Derman and Kani [27], Dupire [32] and Rubinstein [88] have independently shown that an one–factor diffusion model of the form

$$dS = \mu(S,t)\,S\,dt + \sigma(S,t)\,S\,dW_t \tag{5.1.6}$$

for the value of the asset is sufficiently rich to allow a perfect fit to most reasonable volatility smiles [67]. The local volatility function $\sigma(S,t)$ is now a deterministic function dependent on the time as well as on the current asset price level. This model has the advantage that it maintains the completeness and allows for the application of the usual arbitrage–free pricing techniques. Moreover, Dupire has shown in [32] that this local volatility function and hence, the risk–neutral process (5.1.6) is determined when there exists a complete knowledge of arbitrage–free values of European call options (or put options) for all possible strike prices $K > 0$ and maturities $T > 0$.

This group of researchers has tried to extract the true probability distribution of the stock prices from market European option prices by using implied trees, see for instance Rubinstein [88], Dupire [32], Derman and Kani [27], Derman, Kani and Chriss [28].

In this work, we follow a different approach by considering a continuous model for a European call option consistent with the one–factor diffusion model (5.1.6) for the price process of the underlying. Using the common arbitrage–free pricing techniques the resulting European call price model is given by the following deterministic parabolic differential equation with nonconstant coefficients with one of these coefficients containing the volatility function $\sigma(S,t)$ of the diffusion process (5.1.6):

$$\begin{aligned} C_t \;+\; \tfrac{1}{2}\,\sigma^2(S,t)\,S^2\,C_{SS} \;+\; (r(t)-d(t))\,S\,C_S \;&=\; r(t)\,C\,, \quad && (S,t) \in I\!R^+ \times (0\,,T] \\ C(S,T;K,T) \;&=\; \max(S-K,0)\,, && S \in I\!R^+\,. \end{aligned}$$

Some approaches already exist trying to extract this volatility function $\sigma(S,t)$ by calibrating the model European call prices with respect to their market counterparts, see for instance Lagnado and Osher [67], Andersen and Brotherton–Ratcliffe [1], and Coleman and Verma [17]. The formulation of these problems results in inverse problems so that they are naturally ill–posed. Lagnado and Osher [67] overcome this ill–posedness by introducing regularization techniques in their approach. In comparison with it, Coleman and Verma [17] achieve this effect by representing the volatility function $\sigma(S,t)$ by a two–dimensional cubic spline.

Moreover, starting from this continuous deterministic European call price model, Dupire [32] and, in more general form, Andersen and Brotherton–Ratcliffe [1] have shown that there exists an additional model for the European call option which can be thought as being dual to the first one [32]. The first model considers a fixed option, i.e. the strike price K and the maturity T are hold fixed, and gives information about the value of this option for different values S of the underlying asset and times t, i.e. the European call price function presents a function in S and t. On the contrary, in the second model the current asset value S and the time t are hold fixed and the European call price function is variable with strike price K and maturity T. This second model, again, consists in a deterministic parabolic differential equation containing the volatility function σ in one of its coefficients, but this time the volatility function is dependent on the strike price K and the maturity T, i.e. $\sigma(K,T)$ appears in the coefficient of the differential equation:

$$\begin{aligned} C_T \;-\; \tfrac{1}{2}\,\sigma^2(K,T)\,K^2\,C_{KK} \;+\; (r(T)-d(T))\,K\,C_K \;&=\; -\,d(T)\,C\,, \\ & \qquad (K,T) \in I\!R^+ \times (0\,,\bar{T}] \\ C(\sigma(\cdot,\cdot);K,0) \;&=\; \max(S-K,0)\,, \qquad K \in I\!R^+\,. \end{aligned}$$

In accordance with the literature in the following, we will call $\sigma(K,T)$ the instantaneous volatility function, see for instance [32], [1]. Thus, this model contains the volatility values that correspond to European call options with strike price K and maturity T which are the volatility values the financial institutions have an interest in, see the statements given above. Moreover, since in this model the European call price function is variable with strike price K and maturity T a single evaluation of the model establishes the values of European call options for all conceivable strike prices and maturities for fixed S and t. For these reasons, in this work we will use the latter model for constructing the instantaneous volatility function $\sigma(K,T)$ with respect to market European call option, i.e. adapting the instantaneous volatility function $\sigma(K,T)$ so that the model matches the market prices of all European call options in this class.

The mathematical formulation of the construction of the instantaneous volatility function again represents an inverse problem such that in this case we are also confronted with an ill–posed problem. We will apply the Inexact Gauss–Newton Algorithm 2.2.3 for the solution of this problem which overcomes the ill–posedness since using the trust–region strategy for the solution of the Gauss–Newton subproblems enforces a regularization of these problems, see the statements given in Subsection 2.2.2.

5.2 An Extended Black–Scholes Model for European Calls

Before we present the mathematical formulation of the construction problem of the instantaneous volatility function $\sigma(K,T)$, first, we will derive and study the underlying European call option model which gives information about European call prices for all conceivable strike prices K and maturities T for a fixed value S of the underlying asset and time t. In the first part of this section, we will state the major steps given by Dupire [32] and Andersen and Brotherton–Ratcliffe [1], respectively, of the derivation of this European call option model. The model consists in a deterministic second–order parabolic initial–value problem with some coefficients of the parabolic differential equation containing the space variable K. Since this can lead to instabilities when treating the differential equation numerically, in the following subsection, we will carry out a transformation of the differential equation to eliminate the variable K in these coefficients. Finally in the last subsection, we examine the existence and uniqueness of a solution to the transformed parabolic differential equation by falling back on results presented by Friedman in [37].

5.2.1 The “Dual” European Call Price Model

We assume that the price process of the underlying asset fulfils the one–factor diffusion model (5.1.6). Then, by using the usual arbitrage-free pricing techniques, it follows that the price function $C(S,t;K,T)$ of a European call option is given by the solution of the extended Black–Scholes model

$$\begin{aligned} &C_t \,+\, \tfrac{1}{2}\,\sigma^2(S,t)\,S^2\,C_{SS} \,+\, (r(t)-d(t))\,S\,C_S \,=\, r(t)\,C\,, && (S,t)\in I\!R^+\times(0\,,T] \\ &C(S,T;K,T) \,=\, \max(S-K,0)\,, && S\in I\!R^+\,, \end{aligned} \tag{5.2.1}$$

see for instance Avellaneda and Laurence [3]. This model elucidates the dependence of the value of a European call option, $C(S,t;K,T)$, on the value of the underlying, S, the current time, t, the strike price, K, the maturity, T, the instantaneous interest rate, $r(t)$, the continuous dividend rate, $d(t)$, and the volatility function $\sigma(S,t)$. In the extended Black–Scholes model (5.2.1), the European call price function varies with the current value of the underlying S and the time t. Dupire [32] (see also [33]) and, in more general form, Andersen and Brotherton–Ratcliffe [1] have shown that there also exists a European call price model in which the European call price function is given as a function of the strike price K and the maturity T. These two problems can be thought as being dual to each other, however, the relationship is not so universal [32]. In the following, we will briefly quote their derivation for the latter model to elucidate the relation between these two models. We will follow mainly the statements given by Andersen and Brotherton–Ratcliffe given in [1].

Applying the Feynman–Kac Theorem (see for instance Karatzas and Shreve [63, Theorem 5.7.6]) to the solution $C(S,t;K,T)$ of the parabolic initial–value problem (5.2.1) yields that $C(S,t;K,T)$ admits the stochastic representation

$$\begin{aligned} C(S,t;K,T) &= \exp\left(-\textstyle\int_t^T r(\tau)\,d\tau\right) \int\limits_0^\infty \max(u-K,0)\,p(S,t;u,T)\,du \\ &= \exp\left(-\textstyle\int_t^T r(\tau)\,d\tau\right) \int\limits_K^\infty (u-K)\,p(S,t;u,T)\,du \end{aligned} \tag{5.2.2}$$

with $p(S,t;u,T)$ being the risk–neutral transition probability density of the process S determined by (5.1.6) assuming that the coefficients $r(t)\,,d(t)\;:\;[0\,,T]\;\rightarrow\;I\!R$ are continuous, $\sigma(S,t)\,,\mu(S,t)\;:\;I\!R\times[0\,,\infty)\;\rightarrow\;I\!R$ are locally uniform bounded in t, Lipschitz–continuous in S (ensures the existence and uniqueness of a solution S to the one–factor diffusion model (5.1.6), see for instance Irle [60, Theorem 13.4 and 13.5]) and that they satisfy the linear growth condition

$$\|\mu(S,t)\|^2 \;+\; \|\sigma(S,t)\|^2 \le k^2\,(1+\|S\|^2)$$

for some constant $k > 0$. Note that the condition on the initial condition of the Cauchy problem in the Feynman–Kac Theorem, i.e.

$$|f(S)| \;\le\; k\,\left(1+\|S\|^{2\nu}\right) \quad \text{or} \quad f(S)\;\ge\;0\,, \quad \text{for } S\in I\!R^+$$

is satisfied for the end condition given in the end–value problem (5.2.1) and that the end–value problem (5.2.1) can be transformed into an initial–value problem just by transforming the time variable t into the time to maturity $T-t$. The transition probability density $p(S,t;u,T)$ in the stochastic representation (5.2.2) of $C(S,t;K,T)$ satisfies for fixed (S,t) the Kolmogorov forward equation

$$\begin{aligned} &\tfrac{\partial}{\partial T}p + \tfrac{\partial}{\partial u}([r(T)-d(T)]up) - \tfrac{1}{2}\tfrac{\partial^2}{\partial u^2}(\sigma(u,T)^2u^2p) \;=\; 0\,, \quad (u,T)\in I\!R^+\times(t,\bar{T}]\,, \\ &p(S,t;u,t) = \delta(S-u)\,, \qquad\qquad u\in I\!R^+\,, \end{aligned} \tag{5.2.3}$$

where $\delta(\cdot)$ is the Dirac delta–function, see for instance Cox and Miller [20, Chap. 5] or Karatzas and Shreve [63, Chap. 5].

Moreover, differentiation of $C(S,t;K,T)$ in (5.2.2) twice with respect to K yields

$$\begin{aligned} \tfrac{\partial}{\partial K}C(S,t;K,T) &= -\exp\left(-\textstyle\int_t^T r(\tau)\,d\tau\right) \int\limits_K^\infty p(S,t;u,T)\,du \\ \tfrac{\partial^2}{\partial K^2}C(S,t;K,T) &= \exp\left(-\textstyle\int_t^T r(\tau)\,d\tau\right)\,p(S,t;K,T) \end{aligned}$$

and, thus,

$$p(S,t;K,T) = \exp\left(\int_t^T r(\tau)\,d\tau\right)\ \frac{\partial^2}{\partial K^2} C(S,t;K,T)\,. \tag{5.2.4}$$

This last equation makes clear that the risk–neutral transition density $p(S,t;K,T)$ can be recovered directly from market prices when market European call options exist for all conceivable $K > 0$ and $T > 0$, a result which was already was presented by Breeden and Litzenberger in [13].

Using the representation (5.2.4) of the risk–neutral transition density $p(S,t;K,T)$, the different terms in the forward equation (5.2.3) become

$$\begin{aligned}
\frac{\partial}{\partial T} p &= \exp\left(\int_t^T r(\tau)\,d\tau\right)\ \frac{\partial}{\partial T}\left(\frac{\partial^2}{\partial K^2} C\right) + \frac{\partial}{\partial T}\ \exp\left(\int_t^T r(\tau)\,d\tau\right)\ \frac{\partial^2}{\partial K^2} C \\
&= \exp\left(\int_t^T r(\tau)\,d\tau\right)\ \left(\frac{\partial^2}{\partial K^2}\left(\frac{\partial}{\partial T} C\right) + r(T)\,\frac{\partial^2}{\partial K^2} C\right) \\
\frac{\partial}{\partial K}\left([r(T)-d(T)]\,K\,p\right) &= \exp\left(\int_t^T r(\tau)\,d\tau\right)\ (r(T)-d(T))\,\frac{\partial}{\partial K}\left(K\,\frac{\partial^2}{\partial K^2} C\right) \\
\frac{1}{2}\frac{\partial^2}{\partial K^2}\left(\sigma(K,T)^2\,K^2\,p\right) &= \exp\left(\int_t^T r(\tau)\,d\tau\right)\ \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(\sigma(K,T)^2\,K^2\,\frac{\partial^2}{\partial K^2} C\right)
\end{aligned} \tag{5.2.5}$$

and inserting them into (5.2.3) yields

$$\frac{\partial^2}{\partial K^2}\frac{\partial}{\partial T} C + r(T)\,\frac{\partial^2}{\partial K^2} C + (r(T)-d(T))\,\frac{\partial}{\partial K}\left(K\frac{\partial^2}{\partial K^2} C\right) - \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(\sigma^2(K,T)K^2\frac{\partial^2}{\partial K^2} C\right) = 0\,.$$

Integration of this equation twice with respect to K produces

$$\begin{aligned}
\frac{\partial}{\partial T} C + r(T)\,C + (r(T)-d(T))\,\left(K\,\frac{\partial}{\partial K} C - C\right) - \frac{1}{2}\left(\sigma^2(K,T)K^2\frac{\partial^2}{\partial K^2} C\right) \\
= \quad V(T)K + W(T)\,,
\end{aligned} \tag{5.2.6}$$

where V and W are arbitrary functions of time. Assuming that the functions appearing in (5.2.6) show sufficient regularity features to make all terms, that include C, approach zero as K approaches infinity, see also Dupire [33], Andreasen [2], then, the integration functions V and W must be zero. Thus, the price process of a European call option is also described by the parabolic initial–value problem

$$C_T - \frac{1}{2}\sigma(K,T)^2K^2C_{KK} + (r(T)-d(T))KC_K + d(T)C = 0\,, \quad (K,T) \in I\!R^+ \times (0\,,\bar{T}]\,, \tag{5.2.7}$$

subject to the initial condition

$$C(S,t;K,0) = \max(S-K,0)\,, \qquad K \in I\!R^+\,, \tag{5.2.8}$$

however, now the European call price function is dependent on the strike price K and the maturity T.

An important aspect in the derivation of the initial–value problem (5.2.7), (5.2.8) is the representation (5.2.4) of the risk–neutral transition density by the second derivative of the European call price function with respect to the strike price K, which holds because the intrinsic value of a European call happens to be the second integral of a Dirac function [32]. Thus, the initial–value problem (5.2.7), (5.2.8) applies only to European call options whereas by choosing the appropriate initial condition in (5.2.1) the latter model is valid for any contingent claim. Hence, the dual relationship between the problems (5.2.1) and (5.2.7), (5.2.8) holds only for European call options.

Because of the reasons given in the previous section, we proceed working with the initial value problem (5.2.7), (5.2.8).

5.2.2 Transformation of the “Dual” Extended Black–Scholes Equation

Before we further examine the partial differential equation (5.2.7), we eliminate the variable K in the coefficients of the equation. This can be done by changing the variable K to

$$K = S\,\exp(x) \tag{5.2.9}$$

with $S \neq 0$ and setting

$$C(K,T) = S\,c(\ln \tfrac{K}{S}, T) = S\,c(x,T)\,. \tag{5.2.10}$$

Assuming $S \neq 0$ is a natural restriction since in the case of $S = 0$ the stochastic process (5.1.6) elucidates that the asset price will stay there also for future times so that trading contingent claims with this underlying would not make any sense anymore.

Using the transformation (5.2.9) and the definition (5.2.10), the partial derivatives of C with respect to K can be expressed in terms of the transformed European call function c according the following identities (note that $x = \ln \frac{K}{S}$)

$$\begin{aligned}
\tfrac{\partial}{\partial K} C(K,T) &= S\,\tfrac{\partial}{\partial x} c(x,T)\,\tfrac{\partial}{\partial K}(\ln \tfrac{K}{S}) \\
&= \tfrac{S}{K}\,\tfrac{\partial}{\partial x} c(x,T)\,, \\
\tfrac{\partial^2}{\partial K^2} C(K,T) &= -\tfrac{S}{K^2}\,\tfrac{\partial}{\partial x} c(x,T) + \tfrac{S}{K}\,\tfrac{\partial^2}{\partial x^2} c(x,T)\,\tfrac{\partial}{\partial K}(\ln \tfrac{K}{S}) \\
&= \tfrac{S}{K^2}\left(\tfrac{\partial^2}{\partial x^2} c(x,T) - \tfrac{\partial}{\partial x} c(x,T)\right)\,.
\end{aligned} \tag{5.2.11}$$

Replacing the partial derivatives in the differential equation (5.2.7) with the corresponding expressions in (5.2.11) we get the following partial differential equation for the transformed European call function $c(x,T)$ (for clearness, we neglect the arguments (x,T) of c in the following)

$$\begin{aligned}
0 &= S\,\tfrac{\partial}{\partial T} c - \tfrac{1}{2}\sigma^2(K,T)\,K^2 \tfrac{S}{K^2}\left(\tfrac{\partial^2}{\partial x^2} c - \tfrac{\partial}{\partial x} c\right) + (r(T) - d(T))\,K \tfrac{S}{K}\,\tfrac{\partial}{\partial x} c + d(T)\,S\,c \\
&= S\left(\tfrac{\partial}{\partial T} c - \tfrac{1}{2}\hat{\sigma}^2(x,T)\,\tfrac{\partial^2}{\partial x^2} c + \left(r(T) - d(T) + \tfrac{1}{2}\hat{\sigma}^2(x,T)\right)\tfrac{\partial}{\partial x} c + d(T)\,c\right)\,.
\end{aligned}$$

To express the volatility function σ also in terms of the transformed variable x, we have changed slightly the notation for $\sigma(K,T)$ to

$$\hat{\sigma}(x,T) := \sigma(S\,\exp(x)\,, T) = \sigma(K,T)\,.$$

According to the transformation (5.2.9), the initial condition (5.2.8) in terms of c has the form

$$C(K,0) = \max(S - K, 0) = S\,\max(1 - \tfrac{K}{S}, 0) = S\,\max(1 - \exp(x)\,, 0) = S\,c(x,0)\,.$$

Thus, $c(x,T)$ is the solution to the parabolic differential equation

$$\tfrac{\partial}{\partial T}c \;+\; \big(b(T)+\tfrac{1}{2}\hat{\sigma}^2(x,T)\big)\;\tfrac{\partial}{\partial x}c \;-\; \tfrac{1}{2}\hat{\sigma}^2(x,T)\;\tfrac{\partial^2}{\partial x^2}c \;=\; -\,d(T)\,c \tag{5.2.12}$$

subject to the initial condition

$$c(x,0) = \max(1-\exp(x)\,,0)\,, \tag{5.2.13}$$

with

$$b(T) := r(T) - d(T)\,. \tag{5.2.14}$$

In the following, we will examine the transformed parabolic differential equation (5.2.12), (5.2.13) since this problem is more advantageous to the subsequent numerical treatment. We will see later on that the stability of finite difference methods applied to the transformed forward equation (5.2.12) for European call options depends on the discretization size of the x–space. By contrast, the stability of finite difference methods applied to the original parabolic differential equation (5.2.7) depends on the size of the variable K. Hence, the scheme can be stable for small K and unstable for large K, see for instance Wilmott et al [107] and Remark 5.4.16 in this work. Thus, it is not evident whether the scheme is stable and it can happen that negative option values are predicted for large values of K. However, this behavior does not occur with the discretization of the transformed equation (5.2.12) since the resulting system is either stable at every mesh point or unstable at every mesh point.

5.2.3 Existence and Uniqueness of Solutions to the Transformed “Dual” Extended Black–Scholes Equation

In this section, we examine the existence and the uniqueness of the solution to the parabolic differential equation (5.2.12) with initial condition (5.2.13). The existence as well as the uniqueness of solutions to linear parabolic differential equations with nonconstant smooth coefficients is already well estabilished and can be found for instance in the monographs by Friedman [37], Itô [61] or Ladyženskaja, Solonnikov and Ural’ceva [66].

The following statements are the main results of Friedman in [37, Chapter 1] regarding the existence and uniqueness of the solution to the Cauchy problem (5.2.12), (5.2.13). Let

$$\mathcal{L}c \;=\; \tfrac{1}{2}\hat{\sigma}^2(x,T)\,\tfrac{\partial^2}{\partial x^2}c - (b(T)+\tfrac{1}{2}\hat{\sigma}^2(x,T))\,\tfrac{\partial}{\partial x}c - d(T)\,c - \tfrac{\partial}{\partial T}c \tag{5.2.15}$$

for $(x,T)\in I\!\!R\times(0,\bar{T}]$. Then, a solution to the parabolic differential equation $\mathcal{L}c=0$ is defined as

Definition 5.2.1 (See Friedman [37, Chap. 1.1].) The function $c(x,T)$ is a solution to the parabolic differential equation $\mathcal{L}c=0$ in the region $I\!\!R\times(0,\bar{T}]$ if

$$\tfrac{\partial}{\partial x}c(x,T)\,,\quad \tfrac{\partial^2}{\partial x^2}c(x,T)\,,\quad \tfrac{\partial}{\partial T}c(x,T)$$

are continuous functions in $I\!\!R\times(0,\bar{T}]$ and $\mathcal{L}c(x,T)=0$ for each point $(x,T)\in I\!\!R\times(0,\bar{T}]$.

To guarantee the existence and the uniqueness of a solution $c(x, T)$ to the parabolic differential equation $\mathcal{L}c = 0$, the coefficients of the differential operator $\mathcal{L}$ have to satisfy certain conditions. The coefficient of the second–order term in the definition of $\mathcal{L}$ (5.2.15) consists of the instantaneous volatility function $\hat{\sigma}(x, T)$ of the risky asset underlying the European call option. We assume that this riskiness is always existing so that $\hat{\sigma}^2(x, T)$ is bounded away from 0, i.e. there exists a positive constant λ_0 such that $\lambda_0 \leq \hat{\sigma}^2(x, T)$. Additionally, we assume that the riskiness of the underlying asset is bounded above by a positive constant λ_1 such that for $\hat{\sigma}^2(x, T)$ holds

$$\lambda_0 \leq \hat{\sigma}^2(x, T) \leq \lambda_1 \quad \text{for all} \quad (x, T) \in I\!R \times [0, \bar{T}] \,. \tag{5.2.16}$$

Condition (5.2.16) ensures that the differential equation (5.2.12) is uniformly parabolic. Moreover, we assume that $r(T)$ and $d(T)$ are bounded continuous functions in $[0, \bar{T}]$, $\hat{\sigma}^2(x, T)$, $\frac{\partial}{\partial x}\hat{\sigma}^2(x, T)$ and $\frac{\partial^2}{\partial x^2}\hat{\sigma}^2(x, T)$ are bounded continuous functions in $I\!R \times [0, \bar{T}]$, which satisfy

$$\begin{aligned} |\hat{\sigma}^2(x, T) - \hat{\sigma}^2(x_0, T_0)| &\leq \kappa \left(|x - x_0|^q + |T - T_0|^{\frac{q}{2}}\right) \\ |\tfrac{\partial}{\partial x}\hat{\sigma}^2(x, T) - \tfrac{\partial}{\partial x}\hat{\sigma}^2(x_0, T)| &\leq \kappa \, |x - x_0|^q \\ |\tfrac{\partial^2}{\partial x^2}\hat{\sigma}^2(x, T) - \tfrac{\partial^2}{\partial x^2}\hat{\sigma}^2(x_0, T)| &\leq \kappa \, |x - x_0|^q \end{aligned} \tag{5.2.17}$$

for all $(x, T)\,, (x_0, T_0)\,, (x_0, T) \in I\!R \times [0, \bar{T}]$, $q > 0$.

Friedman [37] derived the existence of the solution to a second–order linear parabolic differential equation by using the parametrix method. In this method a fundamental solution $\Gamma(x, T; \chi, \tau)$ of $\mathcal{L}c = 0$ is constructed.

Definition 5.2.2 (See Friedman [37, Chap. 1.1].) A fundamental solution of $\mathcal{L}c = 0$ (in $I\!R \times [0, \bar{T}]$) is a function $\Gamma(x, T; \chi, \tau)$ defined for all $(x, T)\,, (\chi, \tau) \in I\!R \times [0, \bar{T}],\ T > \tau$ which satisfies the following conditions

1. for fixed $(\chi, \tau) \in I\!R \times [0, \bar{T}]$ it satisfies, as a function of $(x, T),\ x \in I\!R,\ \tau < T < \bar{T}$, the equation
$$\tfrac{1}{2}\hat{\sigma}^2(x, T)\, \tfrac{\partial^2}{\partial x^2}c - \left(b(T) + \tfrac{1}{2}\hat{\sigma}^2(x, T)\right) \tfrac{\partial}{\partial x}c - d(T)\, c - \tfrac{\partial}{\partial T}c \;=\; 0$$
2. for every continuous function $f(x)$ in $I\!R$ with
$$|f(x)| \;\leq\; p_1 \, \exp(h\, x^2)\,,$$
if $x \in I\!R$, then
$$\lim_{T \to \tau} \int\limits_{-\infty}^{\infty} \Gamma(x, T; \chi, \tau)\, f(\chi)\, d\chi = f(x)$$

According to Friedman [37], a fundamental solution $\Gamma(x, T; \chi, \tau)$ of $\mathcal{L}c = 0$ (in $I\!R \times [0, \bar{T}]$) is given by

$$\Gamma(x, T; \chi, \tau) \;=\; Z(x, T; \chi, \tau) \;+\; \int\limits_{\tau}^{T} \int\limits_{-\infty}^{\infty} Z(x, T; \eta, \upsilon)\; \Phi(\eta, \upsilon; \chi, \tau)\; d\eta\, d\upsilon \tag{5.2.18}$$

where $Z(x,T;\chi,\tau)$ is a fundamental solution of the following differential equation without minor terms and frozen leading coefficient

$$\tfrac{1}{2}\hat{\sigma}^2(\chi,\tau)\,\tfrac{\partial^2}{\partial x^2}c \;-\; \tfrac{\partial}{\partial T}c \;=\; 0\,.$$

It is well–known that the fundamental solution $Z(x,T;\chi,\tau)$ of the latter differential equation is given by the so–called Gaussian kernel

$$Z(x,T;\chi,\tau) = \frac{1}{\sqrt{2\pi\,(T-\tau)}\,\hat{\sigma}(\chi,\tau)}\;\exp\left(\frac{(x-\chi)^2}{2\,\hat{\sigma}^2(\chi,\tau)\,(T-\tau)}\right)\,.$$

Furthermore, the density $\Phi(\eta,\upsilon;\chi,\tau)$ in the integral term of (5.2.18) is determined by the Volterra integral equation

$$\Phi(x,T;\chi,\tau) \;=\; \mathcal{L}Z(x,T;\chi,\tau) \;+\; \int\limits_{\tau}^{T}\int\limits_{-\infty}^{\infty} \mathcal{L}Z(x,T;\eta,\upsilon)\;\Phi(\eta,\upsilon;\chi,\tau)\;d\eta\,d\upsilon\,.$$

Thus, the solution to the parabolic differential equation (5.2.12) with initial condition (5.2.13) is given by

$$c(x,T) \;=\; \int\limits_{-\infty}^{\infty} \Gamma(x,T;\chi,0)\;c(\chi,0)\;d\chi \tag{5.2.19}$$

if the initial condition (5.2.13) satisfies

$$|c(x,0)| \;\le\; p_2\;\exp(h\,x^2) \tag{5.2.20}$$

where h is any positive constant satisfying

$$h \;<\; \frac{1}{4\,\bar{T}\,\lambda_1}\,, \tag{5.2.21}$$

and $c(x,T)$ satisfies

$$|c(x,T)| \;\le\; p_3\;\exp(k\,x^2) \quad \text{for} \quad (x,T)\in I\!R\times[0,\bar{T}] \tag{5.2.22}$$

with p_3 is a positive constant and k is a constant depending only on h, λ_1 and $\bar{T}$, see Friedman [37, Chap. 1, Theorem 12].

Furthermore, Friedman [37, Chap. 1, Theorem 16] showed that there exists at most one solution to the parabolic differential equation (5.2.12) with initial condition (5.2.13) that satisfies the boundedness condition

$$\int\limits_{0}^{\bar{T}}\int\limits_{-\infty}^{\infty} |c(x,T)|\;\exp(-k\,x^2)\,dx\,dT \;<\; \infty$$

for some positive constant k.

Theorem 5.2.3 *Suppose* $r(T)$ *and* $d(T)$ *are bounded continuous functions in* $[0,\bar{T}]$, $\hat{\sigma}^2(x,T)$, $\frac{\partial}{\partial x}\hat{\sigma}^2(x,T)$ *and* $\frac{\partial^2}{\partial x^2}\hat{\sigma}^2(x,T)$ *are bounded continuous functions in* $\mathbb{R} \times [0,\bar{T}]$ *and the conditions* (5.2.16) *and* (5.2.17) *are satisfied. Then, the solution* (5.2.19) *of the parabolic initial–value problem* (5.2.12), (5.2.13) *that is unique with respect to the set*

$$\mathcal{B} = \{c(x,T) \; : \; \int_0^{\bar{T}} \int_{-\infty}^{\infty} |c(x,T)| \, \exp(-k\,x^2)\, dx\, dT \; < \; \infty \;\; \textit{for some} \;\; k > 0\}$$

represents the price function of a European call option.

Proof: To ensure the existence of the solution (5.2.19) to the parabolic initial–value problem (5.2.12), (5.2.13) we have to verify that the initial condition (5.2.13) given by

$$\begin{aligned} c(x,0) \;&=\; \max(1 - \exp(x)\,, 0) \\ &=\; \begin{cases} 1 - \exp(x) & , \quad x < 0 \\ \quad 0 & , \quad x \geq 0 \end{cases} \end{aligned}$$

satisfies condition (5.2.20).

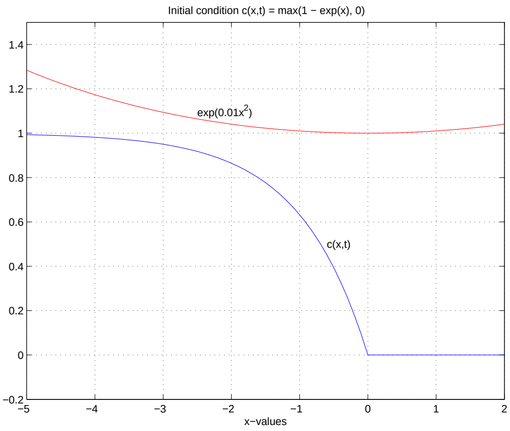


FIGURE 5.1: Boundedness of the initial condition $c(x,0)$

For all $x \geq 0$ the initial condition (5.2.13) is equal to 0 such that in this case the inequality (5.2.20) is satisfied for all positive constants p_2 and h. Furthermore, the function $c(x,0)$ is positive and strictly decreasing for $x < 0$ and $1 - \exp(x)$ tends to 1 for x tending to $-\infty$. Hence, we can choose an arbitrary positive constant h to ensure the condition (5.2.20) if only $p_2 \geq 1$ (see also Figure 5.1). Thus, the existence of the solution (5.2.19) to the differential equation (5.2.12) with initial condition (5.2.13) is ensured for any arbitrary time interval $[0,\bar{T}]$ with $\bar{T} < \infty$.

Moreover, it follows with (5.2.22) that $c(x,T)$ in (5.2.19) is the unique solution in the set

$$\mathcal{B} = \{c(x,T) \; : \; \int_0^{\bar{T}} \int_{-\infty}^{\infty} |c(x,T)| \, \exp(-k\,x^2)\, dx\, dT \; < \; \infty \;\; \text{for some} \;\; k > 0\}$$

to the differential equation (5.2.12), (5.2.13).

To show that the solution (5.2.19) represents the price function of a European call option, we only have to verify that the European call price function belongs to the set $\mathcal{B}$. Merton [73] showed that from no–arbitrage arguments follows that the price of a European call option has to satisfy

$$C(K,T) \;<\; S \quad \text{for all} \quad K \geq 0 \quad \text{and} \quad T \in [0, \bar{T}]\,.$$

and thus, because of (5.2.10) $c(x,T)$ has to fulfil

$$c(x,T) \;<\; 1 \quad \text{for all} \quad x \in I\!R \quad \text{and} \quad T \in [0, \bar{T}]\,.$$

Thus, the transformed European call price function belongs to the set $\mathcal{B}$ such that Theorem 5.2.3 is proven.

■

5.3 Determination of the Instantaneous Volatility Function

After establishing the European call option model, we now return to the construction of the instantaneous volatility function $\sigma(K,T)$ respective $\hat{\sigma}(x,T)$.

The previous section has shown that if the functional form of $\hat{\sigma}(x,T)$ is specified and given the value S of the underlying, the risk–free instantaneous interest rate $r(T)$ and the continuous dividend rate $d(T)$, then a European call option with transformed strike price $x = \ln(\frac{K}{S})$ and maturity T has a unique value $c(\hat{\sigma}(\cdot,\cdot); x, T)$. Note that we have changed slightly the notation to express the dependence of the European option price on the particular choice of the instantaneous volatility function $\hat{\sigma}(x,T)$.

Now, suppose that we are given m market prices of European call options with strike prices K_l and maturities T_l, $l = 1, \ldots, m$. Let the bid and ask prices observed at $t = 0$ of a European call option with strike price K_l and maturity T_l be denoted by $\hat{C}^b_{K_l,T_l}$ and $\hat{C}^a_{K_l,T_l}$, respectively, $l = 1, \ldots, m$. In this context, constructing the instantaneous volatility function $\hat{\sigma}(x,T)$ with respect to market European call options means that an instantaneous volatility function $\hat{\sigma}(x,T)$ has to be found such that the solution to the initial–value problem (5.2.12), (5.2.13) evaluated at $x_l = \ln(\frac{K_l}{S})$ and T_l falls between the corresponding bid and ask quotes, i.e.

$$\hat{C}^b_{K_l,T_l} \;\leq\; S\, c(\hat{\sigma}(\cdot,\cdot); x_l, T_l) \;\leq\; \hat{C}^a_{K_l,T_l}$$

for $l = 1, \ldots, m$. Since it is somewhat unwieldy working with these inequalities we will alternatively minimize the distance between the model European call values $c(\hat{\sigma}(\cdot,\cdot); x_l, T_l)$ and the mean $\hat{C}_{K_l,T_l} = (\hat{C}^b_{K_l,T_l} + \hat{C}^a_{K_l,T_l})/2$ of the bid and ask quotes, $l = 1, \ldots, m$, with respect to the Euclidean norm. Moreover, existence of an unique solution to the initial–value problem (5.2.12), (5.2.13) is guaranteed if the instantaneous volatility function $\hat{\sigma}(x,t)$ belongs to the set

$$\mathcal{H} \;=\; \{\hat{\sigma}(x,T) \;:\; \hat{\sigma}^2\,, \tfrac{\partial}{\partial x}\hat{\sigma}^2\,, \tfrac{\partial^2}{\partial x^2}\hat{\sigma}^2 \;\text{ bounded, continuous and satsify (5.2.17)}\}\,. \tag{5.3.1}$$

Thus, we alternatively solve the nonlinear least–squares problem

$$\begin{aligned} &\min \ \tfrac{1}{2} \sum_{l=1}^{m} \|c(\hat\sigma(\cdot,\cdot); x_l, T_l) - \hat C_{K_l,T_l}\|^2 \\ &\text{s.t.} \quad \hat\sigma(x,T) \in \mathcal{H} \end{aligned} \tag{5.3.2}$$

with $c(\hat\sigma(\cdot,\cdot); x, T)$ being the solution to the initial–value problem

$$\begin{aligned} c_T + (b(T) + \tfrac{1}{2}\hat\sigma^2(x,T))c_x - \tfrac{1}{2}\hat\sigma^2(x,T)c_{xx} \ &= \ -d(T)c\,, \quad (x,T) \in I\!R \times (0\,,\bar T] \\ c(\hat\sigma(\cdot,\cdot); x, t) \ &= \ \max(1 - e^x, 0)\,, \qquad\qquad\qquad x \in I\!R\,, \end{aligned} \tag{5.3.3}$$

with $b(T)$ defined in (5.2.14).

Hence, the problem of constructing the volatility function $\hat\sigma(x,T)$ presents an inverse problem which is naturally ill–posed. To overcome the ill–posedness, often, regularization terms are imposed into the objective function in (5.3.2), see for instance Lagnado and Osher [67]. Since smoothness of the instantaneous volatility function $\hat\sigma(x,T)$ is important for the existence of a solution of the initial–value problem (see above), they use smoothness as a regularization condition to approximate the volatility function, i.e. their regularized optimization problem is of the form

$$\min_{\hat\sigma(x,T)\in\mathcal{H}} \ \tfrac{1}{2} \sum_{l=1}^{m} \|c(\hat\sigma(\cdot,\cdot); x_l, T_l) - \hat C_{K_l,T_l}\|^2 + \lambda \|\nabla\hat\sigma(x,T)\|_2$$

with λ being a positive constant. In this regularized problem, the size of the regularization parameter λ is responsible for the degree of minimizing the first order derivative of the volatility function, i.e. the degree of smoothness of the volatility function. However, the choice of this regularization parameter appears to be not easy since a large regularization parameter slows down the optimization process while a small regularization parameter might not overcome the ill–posedness of the problem.

In this work, we use a different approach. Using finite difference methods to discretize the initial–value problem (5.3.3) and because of the limited number of observation data, the finite dimensional counterpart of (5.3.2) represents an underdetermined nonlinear least–squares problem, see the following Section 5.4. Using the Inexact Gauss–Newton Algorithm 2.2.3 for the solution of this problem, we overcome the ill–posedness of the nonlinear least–squares problem because of the integrated trust–region strategy. This latter method enforces a regularization of the linear Gauss–Newton subproblems. The regularization with respect to the smoothness of the volatility function will be achieved by scaling the trust–region with respect to a modified discretized version of the differentiation operator.

5.4 Determination of the Discrete Instantaneous Volatility Function

For a numerical treatment of the optimization problem (5.3.2) it is, first of all, necessary to evaluate numerically the theoretical European call price function $c(\hat\sigma(\cdot,\cdot); x, T)$ that is described by the parabolic differential equation (5.3.3). We will solve this differential equation by using a finite difference scheme. Note that the parabolic differential equation (5.3.3) is a pure initial–value problem. Since in practice only a finite number of discretization points regarding the space variable x can be treated in Subsection 5.4.1, we introduce boundary conditions for $c(\hat{\sigma}(\cdot,\cdot);x_{\min},T)$ and $c(\hat{\sigma}(\cdot,\cdot);x_{\max},T)$ with $x_{\min}$ and $x_{\max}$ are chosen sufficiently small and large, respectively. Using these boundary conditions, we will derive a finite difference scheme for the resulting initial–boundary–value problem in Subsection 5.4.2. After examining the well–posedness and the convergence of the finite difference scheme (see Subsection 5.4.3), we will derive the finite dimensional optimization problem that determines the instantaneous volatility function at the grid points of the mesh chosen in the finite difference method. The resulting optimization problem is an underdetermined nonlinear least–squares problem that will be solved with the Inexact Gauss–Newton Algorithm 2.2.3. The ill–posedness of the problem mentioned in the previous section is carried over to the finite dimensional problem. However, the trust–region strategy used in Algorithm 2.2.3 enforces a regularization of the Gauss–Newton subproblems such that these problems will become well–posed. We will carry out the trust–region strategy with respect to the weighted norm $\|\cdot\|_{\bar{D}}$ where $\bar{D}$ is a slight modification of the discrete version of the differentiation operator since we required the solution $\hat{\sigma}(x,T)$ to be sufficiently smooth, i.e. $\hat{\sigma}(x,T)\in\mathcal{H}$.

5.4.1 Derivation of Boundary Conditions

In Section 5.2.3, we have seen that the European call price is already uniquely described by the parabolic initial–value problem (5.3.3). For a numerical determination of the European call price function $c(\hat{\sigma}(\cdot,\cdot);x,T)$, however, we can only consider a finite dimensional number of discretization points regarding the state variable x. Thus, we have to choose a lower bound $x_{\min}$ and an upper bound $x_{\max}$ for the state variable x so that the resulting interval $[x_{\min}\,,x_{\max}]$ contains the region relevant for the construction of the instantaneous volatility function, i.e. $x_{\min}$ and $x_{\max}$ have to be chosen sufficiently small and large, respectively. Hence, we are approximating $I\!R\times[0\,,\bar{T}]$ by the bounded domain $[x_{\min}\,,x_{\max}]\times[0\,,\bar{T}]$ and the pure initial–value problem becomes a initial–boundary–value problem. Therefore, we need to derive wise values of $c(\hat{\sigma}(\cdot,\cdot);x,T)$ at the lateral boundary of $[x_{\min}\,,x_{\max}]\times[0\,,\bar{T}]$, i.e. we need information about the boundary values $c(\hat{\sigma}(\cdot,\cdot);x_{\min},T)$ and $c(\hat{\sigma}(\cdot,\cdot);x_{\max},T)$ for $T\in[0\,,\bar{T}]$.

We first examine the value of $c(\hat{\sigma}(\cdot,\cdot);x,T)$ at the left boundary, i.e. when x is small. Suppose x tends to $-\infty$. Because of the change of variables (5.2.9), this is equivalent to assuming that the strike price K is very small, i.e. K is tending to 0. In this case, the holder of the European call option gets the underlying asset almost for free at the end of the period T. If we suppose that the volatility is constant, then with (5.1.5) it follows that in this case the European call price for $K=0$ is given by

$$C(\sigma(\cdot,\cdot);0,T)=S\,\exp(-\int_0^T d(\tau)\,d\tau)\,,\quad T\in[0\,,\bar{T}]\,,\tag{5.4.1}$$

with $d(T)\,,\ T\in[0\,,\bar{T}]$ is the known dividend yield. This equality also holds for the nonconstant volatility case. We will derive the value of $C(\sigma(\cdot,\cdot);0,T)$ by building an equivalent portfolio that generates the same payout at the expiration date T as the European call option. Suppose first that no dividends are paid on the asset underlying the option, i.e. $d(T) \equiv 0$ for $T \in [0\,,\bar{T}]$. Since the exercise price K is equal to zero, the payout of the option at the expiration date is equivalent to the value of the underlying asset at the expiration date. Hence, the option can be replicated by using a portfolio which consists of one share of the underlying asset and is held constant over the life of the option. Thus, to avoid possibilities of arbitrage the value of the option and the value of this portfolio at the current time must be the same if no payments on either of these assets are made during the life of the option, i.e. the value of the option is equal to the current price of the asset underlying the option. If this relationship is not true, an arbitrageur can easily make a riskless profit by buying the option and selling the stock or vice versa.

Now, suppose that dividends are paid on the underlying asset. The payment of a dividend causes the stock to drop by an amount equal to the dividend. Supposing that a continuous dividend yield of rate d is paid and the stock price rises from S today to S_T at time T, it follows that without dividend payments the stock would have risen from $S\,\exp(-d(T-t))$ today up to S_T at time T. Thus, if the underlying asset of a European call option pays a continuous dividend yield of rate d then the European call option has the same value than the corresponding European call option with the same underlying asset paying no dividends and having the value $S\,\exp(-d(T-t))$ today since the final value of the asset is the same in both cases. Therefore, for a valuation of a European call option with an underlying asset paying a continuous dividend yield of d we can base the calculations on the current value of the asset of $S\,\exp(-d(T-t))$, see for instance Hull [58]. Hence, assuming a time varying dividend yield $d(T)$, $T \in [0\,,\bar{T}]$, (5.4.1) presents the value of a European call option with strike price $K = 0$ and an underlying asset with value S that pays a dividend yield of rate $d(T)\,,\ T \in [0\,,\bar{T}]$.

Since we carry out the numerical computation of the value of the European call option on a bounded domain with respect to the space variable x, we are interested in the value of the European call option for a sufficiently small, but finite, left boundary $x_{\min} = \ln(\frac{K_{\min}}{S})$. In this case, the holder of the European call option still has to pay a small strike price $K_{\min}$ at maturity. Considering the time value of this payment, the value of the European call option at the left boundary $K_{\min} = S\exp(x_{\min})$ becomes

$$C(\sigma(\cdot,\cdot);K_{\min},T) = S\,\exp(-\int_0^T d(\tau)\,d\tau)\;-\;K_{\min}\,\exp(-\int_0^T r(\tau)\,d\tau)\,, \tag{5.4.2}$$

for $T \in [0\,,\bar{T}]$ and with $r(T)\,,\ T \in [0\,,\bar{T}]$, being the riskless interest rate.

For the right boundary of the domain $[x_{\min}\,, x_{\max}]$, the following observation can be made for the value of a European call option. As the strike price increases without bound, it becomes ever more likely that the option will not be exercised, i.e. it becomes ever more likely that at the expiration date the payoff is zero. Thus, the call option is worthless for $K \to \infty$ even if there is a long time to expiration. Hence,

$$\lim_{K\to\infty} C(\sigma(\cdot,\cdot);K,T) = 0\,,\quad T \in [0\,,\bar{T}]\,, \tag{5.4.3}$$

such that the condition for the right boundary $K_{\max} = S\exp(x_{\max})$ is given by

$$C(\sigma(\cdot,\cdot);K_{\max},T) = 0\,,\quad T \in [0\,,\bar{T}]\,. \tag{5.4.4}$$

Note that the boundary conditions (5.4.2) and (5.4.4) ensure the compatibility of the initial and boundary condition of the initial–boundary–value problem.

The transformation (5.2.10) from $C(K,T)$ to $c(x,T)$ and the boundary conditions (5.4.2) and (5.4.4) for the original parabolic differential equation (5.2.7) lead to the boundary condition for the transformed parabolic differential equation (5.2.12):

$$\begin{aligned} c(\hat\sigma(\cdot,\cdot);x_{\min},T) &= \exp(-\int\limits_0^T d(\tau)\,d\tau) - \exp(x_{\min})\exp(-\int\limits_0^T r(\tau)\,d\tau)\,, \\ c(\hat\sigma(\cdot,\cdot);x_{\max},T) &= 0\,, \end{aligned} \tag{5.4.5}$$

for $T \in [0\,,\bar T]$.

5.4.2 Discretization Scheme for the Transformed Extended Black–Scholes Model

For the numerical computation of the solution $c(\hat\sigma(\cdot,\cdot);x,T)$ of the second–order parabolic initial–boundary–value problem consisting of the parabolic differential equation (5.2.12), the initial condition (5.2.13) and the boundary conditions (5.4.5), i.e.

$$\begin{aligned} &\tfrac{1}{2}\,\hat\sigma^2(x,T)\,c_{xx} - \big(b(T) + \tfrac{1}{2}\,\hat\sigma^2(x,T)\big)\;c_x - d(T)\,c = c_T\,, \\ &\qquad\qquad\qquad\qquad\qquad\qquad (x,T) \in (x_{\min}\,,x_{\max}) \times (0\,,\bar T]\,, \\ c(\hat\sigma(\cdot,\cdot);x,0) &= \max(1-\exp(x),0)\,, \qquad x \in [x_{\min}\,,x_{\max}]\,, \\ c(\hat\sigma(\cdot,\cdot);x_{\min},T) &= \exp(-\int\limits_0^T d(\tau)\,d\tau) - \\ &\qquad \exp(x_{\min})\exp(-\int\limits_0^T r(\tau)\,d\tau)\,, \quad T \in [0,\bar T]\,, \\ c(\hat\sigma(\cdot,\cdot);x_{\max},T) &= 0\,, \qquad T \in [0,\bar T]\,, \end{aligned} \tag{5.4.6}$$

we use a finite difference method (see for instance Gustafsson, Kreiss, Oliger [49], Smith [96], Thomas [101]). These methods reduce the continuous partial differential equation (5.2.12) to a discrete set of difference equations by replacing the partial derivatives occurring in the partial differential equation by finite difference approximations. For notation, we neglect the dependence of the European call price function c on the volatility parameter $\hat\sigma(\cdot,\cdot)$ in this subsection by not considering the volatility parameter in the argument of the European call function c, i.e. in this subsection we use the notation $c(x,T)$ for the European call price function. For the discretization of the parabolic differential equation (5.2.12) we especially use the forward difference quotient for the approximation of $c_T(x,T)$, i.e.

$$c_T(x,T) \;\approx\; \frac{c(x,T+\Delta T) - c(x,T)}{\Delta T}\,, \quad \Delta T \text{ small}\,, \tag{5.4.7}$$

the central difference quotient for approximating $c_x(x,T)$, i.e.

$$c_x(x,T) \;\approx\; \frac{c(x+\Delta x,T) - c(x-\Delta x,T)}{2\,\Delta x}\,, \quad \Delta x \text{ small}\,, \tag{5.4.8}$$

and the symmetric central difference quotient for approximating $c_{xx}(x,T)$, i.e.

$$c_{xx}(x,T) \;\approx\; \frac{c(x+\Delta x,T) - 2\,c(x,T) + c(x-\Delta x,T)}{(\Delta x)^2}\,, \quad \Delta x \text{ small}\,. \tag{5.4.9}$$

From Taylor's theorem it immediately follows that the forward difference quotient (5.4.7) is accurate of order $\mathcal{O}(\Delta T)$, i.e.

$$c_T(x,T) \;=\; \frac{c(x,T+\Delta T) - c(x,T)}{\Delta T} \;+\; \mathcal{O}(\Delta T)\,, \tag{5.4.10}$$

if the solution c to the parabolic initial–boundary–value problem (5.4.6) is sufficiently smooth. Furthermore, the central difference quotient (5.4.8) and the symmetric central difference quotient (5.4.9) give approximations of order $\mathcal{O}((\Delta x)^2)$, i.e.

$$c_x(x,T) \;=\; \frac{c(x+\Delta x,T) - c(x-\Delta x,T)}{2\,\Delta x} \;+\; \mathcal{O}((\Delta x)^2)\,, \tag{5.4.11}$$

$$c_{xx}(x,T) \;=\; \frac{c(x+\Delta x,T) - 2\,c(x,T) + c(x-\Delta x,T)}{(\Delta x)^2} \;+\; \mathcal{O}((\Delta x)^2)\,, \tag{5.4.12}$$

again, supposing a sufficient smoothness of the solution c.

Approximating the partial derivatives occurring in the partial differential equation (5.2.12) by using finite difference quotients, the continuous domain $[x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$ can be replaced by a discrete set of grid points [48]. A division of the domain $[x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$ into a mesh is obtained by dividing the x–axis as well as the T–axis into equally spaced nodes a distance Δx and ΔT, respectively, apart, see Figure 5.2. It is also possible to take non–constant space–steps Δx and time–steps ΔT, however, for convenience we confine our examinations on uniform grids. Suppose that the discretization of the time and the spatial axis are chosen so that

$$\bar{T} \;=\; M\,\Delta T \quad \text{and} \quad x_{\max} - x_{\min} \;=\; N\,\Delta x\,, \quad M\,,N \in I\!N\,, \quad N \text{ even}\,. \tag{5.4.13}$$

Moreover, we set for the grid points of the spatial and the time domain

$$x_i \;:=\; x_{\min} + i\,\Delta x \quad \text{and} \quad T_j \;:=\; j\,\Delta T\,, \qquad i = 0\,,\ldots,N\,,\; j = 0\,,\ldots,M\,.$$

Since in the following we are only interested in values of c at these mesh points we use the notation

$$c_{i,j} = c(x_i\,,T_j) \tag{5.4.14}$$

for the exact solution c of the parabolic initial–boundary–value problem (5.4.6) evaluated at the grid point $(x_i\,,T_j)$, $i = 0\,,\ldots,N\,,\; j = 0\,,\ldots,M$. Furthermore, for the coefficients occurring in

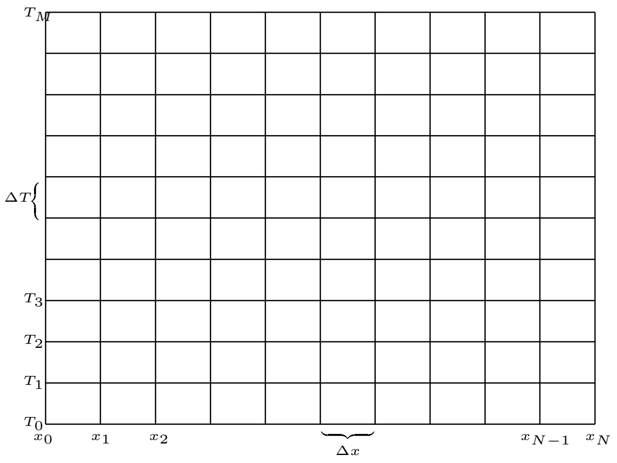


FIGURE 5.2: Discretization of the domain $[x_{\min}\,, x_{\max}] \times [0\,, \bar{T}]$

(5.2.12) we set

$$\begin{aligned} \hat{\sigma}_{i,j} &= \hat{\sigma}(x_i\,, T_j)\,, \\ b_j &= b(T_j)\,, \\ r_j &= r(T_j)\,, \\ d_j &= d(T_j)\,. \end{aligned}$$

for $i = 0\,, \ldots, N\,,\ j = 0\,, \ldots, M$.

We first examine the discretization of the elliptic part of the parabolic differential equation (5.2.12), i.e.

$$\begin{aligned} L\,c(x,T) &= \tfrac{1}{2}\,\hat{\sigma}^2(x,T)\,c_{xx}(x,T) - \\ &\qquad \big(b(T) + \tfrac{1}{2}\,\hat{\sigma}^2(x,T)\big)\ c_x(x,T) - d(T)\,c(x,T)\,. \end{aligned} \tag{5.4.15}$$

Using the central difference quotient (5.4.11) for $c_x(x,T)$ and the symmetric central difference quotient (5.4.12) for $c_{xx}(x,T)$, the elliptic part (5.4.15) of the parabolic differential equation (5.2.12) becomes

$$\begin{aligned} L\,c(x_i\,,T_j) &= \tfrac{1}{2}\hat{\sigma}^2_{i,j}\ \frac{c_{i+1,j} - 2\,c_{i,j} + c_{i-1,j}}{(\Delta x)^2} - \\ &\qquad \big(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\big)\ \frac{c_{i+1,j} - c_{i-1,j}}{2\,\Delta x}\ -\ d_j\,c_{i,j} + \mathcal{O}((\Delta x)^2)\,, \end{aligned}$$

$i = 1\,, \ldots, N-1\,,\ j = 0\,, \ldots, M$, where we have restricted our attention to the values of c on the discrete mesh. Ignoring the term $\mathcal{O}((\Delta x)^2)$, an approximation of the elliptic part $L\,c(x_i\,,T_j)$ is given by

$$\begin{aligned} \tilde{L}_{i,j}\,z_{i,j} = \tfrac{1}{2}\hat{\sigma}^2_{i,j}\ &\frac{z_{i+1,j} - 2z_{i,j} + z_{i-1,j}}{(\Delta x)^2} - \\ &\big(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\big)\ \frac{z_{i+1,j} - z_{i-1,j}}{2\Delta x} - d_j\,z_{i,j}\,, \end{aligned} \tag{5.4.16}$$

$i = 1, \dots, N-1,\ j = 0, \dots, M$, where the use of $z_{i,j}$ emphasizes that the scheme (5.4.16) is only an approximation of the elliptic differential equation L in (5.4.15). Note that $z_{0,j}$ and $z_{N,j}$ in (5.4.16) are given by the boundary conditions (5.4.5) of the initial–boundary–value problem (5.4.6), i.e.

$$\begin{aligned} z_{0,j} &= \exp(-\int_0^{T_j} d(\tau)\, d\tau) \ - \ \exp(x_0) \exp(-\int_0^{T_j} r(\tau)\, d\tau)\,, \\ z_{N,j} &= 0\,. \end{aligned} \tag{5.4.17}$$

A rearrangement of the terms in the right hand side of (5.4.16) yields

$$\tilde{L}_{i,j}\, z_{i,j} = -\tfrac{1}{(\Delta x)^2} \left(v_i^j\, z_{i-1,j} + m_i^j\, z_{i,j} + u_i^j\, z_{i+1,j} \right)\,, \tag{5.4.18}$$

for $i = 1, \dots, N-1,\ j = 0, \dots, M$, where

$$\begin{aligned} v_i^j &= -\tfrac{1}{2} \left(\hat{\sigma}_{i,j}^2 + \Delta x \left(b_j + \tfrac{1}{2} \hat{\sigma}_{i,j}^2 \right) \right)\,, \\ m_i^j &= \hat{\sigma}_{i,j}^2 + (\Delta x)^2\, d_j\,, \\ u_i^j &= -\tfrac{1}{2} \left(\hat{\sigma}_{i,j}^2 - \Delta x \left(b_j + \tfrac{1}{2} \hat{\sigma}_{i,j}^2 \right) \right)\,, \end{aligned} \tag{5.4.19}$$

for $j = 0, \dots, M$. For the numerical computation of the integral terms $\int_0^{T_j} d(\tau)\, d\tau$ and $\int_0^{T_j} r(\tau)\, d\tau$ occurring in the first boundary condition in (5.4.17), we use the trapezoidal rule, i.e. we approximate the integrals $\int_0^{T_j} d(\tau)\, d\tau$ and $\int_0^{T_j} r(\tau)\, d\tau$ by

$$\int_0^{T_j} d(\tau)\, d\tau \ \approx \ \tfrac{\Delta T}{2} \sum_{k=0}^{j-1} (d_k + d_{k+1}) \quad \text{and} \quad \int_0^{T_j} r(\tau)\, d\tau \ \approx \ \tfrac{\Delta T}{2} \sum_{k=0}^{j-1} (r_k + r_{k+1})\,. \tag{5.4.20}$$

This approximation is accurate of order $\mathcal{O}((\Delta T)^2)$, see for instance Stoer [99].

Now, using matrix notation, equation (5.4.16) can be rewritten in a more compact form

$$\tilde{L}_j z_j = -\left(\bar{D}_j z_j + \bar{B}_j \right)\,, \quad j = 0, \dots, M\,,$$

where

$$z_j = (z_{1,j}, \dots, z_{N-1,j}) \in I\!R^{N-1}\,, \tag{5.4.21}$$

and $\bar{D}_j$ is a $(N-1) \times (N-1)$–dimensional tridiagonal matrix

$$\bar{D}_j \ = \ \tfrac{1}{(\Delta x)^2}\, D_j \in I\!R^{(N-1)\times(N-1)} \tag{5.4.22}$$

with D_j containing the coefficients m_i^j, u_i^j and v_i^j in (5.4.19) on the diagonal and on the super- and subdiagonal, i.e.

$$D_j = \begin{bmatrix} m_1^j & u_1^j & & & \\ v_2^j & m_2^j & u_2^j & & \\ & \ddots & \ddots & \ddots & \\ & & v_{N-2}^j & m_{N-2}^j & u_{N-2}^j \\ & & & v_{N-1}^j & m_{N-1}^j \end{bmatrix} \in I\!R^{(N-1)\times(N-1)}\,, \tag{5.4.23}$$

$j = 0, \dots, M$. The vector $\bar{B}_j$ at the extreme right–hand side of this equation arises from the boundary conditions (5.4.17) in combination with (5.4.20), i.e. $\bar{B}_j$ contains the values $z_{0,j}$ and $z_{N,j}$

$$\bar{B}_j = \tfrac{1}{(\Delta x)^2} \left(v_1^j \, z_{0,j} \, , 0 \, , \dots , 0 \, , u_{N-1}^j \, z_{N,j} \right)^T \in I\!R^{N-1} \tag{5.4.24}$$

where $z_{0,j}$ and $z_{N,j}$ are given by, $j = 0, \dots, M$,

$$\begin{aligned} z_{0,j} &= \exp(-\tfrac{\Delta T}{2} \sum_{k=0}^{j-1} (d_k + d_{k+1})) - \exp(x_0) \, \exp(-\tfrac{\Delta T}{2} \sum_{k=0}^{j-1} (r_k + r_{k+1})) \, , \\ z_{N,j} &= 0 \, . \end{aligned} \tag{5.4.25}$$

Lemma 5.4.1 *The elliptic part of the parabolic differential equation* (5.2.12)*, i.e.*

$$L \, c(x,T) = \tfrac{1}{2} \, \hat{\sigma}^2(x,T) \, c_{xx}(x,T) - \left(b(T) + \tfrac{1}{2} \, \hat{\sigma}^2(x,T) \right) \, c_x(x,T) - d(T) \, c(x,T)$$

in discretized form is given by

$$\tilde{L}_j z_j = - \left(\bar{D}_j z_j + \bar{B}_j \right) \, , \quad j = 0, \dots, M \, ,$$

with z_j , $\bar{D}_j$ and $\bar{B}_j$ defined in (5.4.21), (5.4.22)*, and* (5.4.24)*, respectively, and by using the central difference quotient* (5.4.8) *and the symmetric central difference quotient* (5.4.9) *for approximating c_x and c_{xx}, respectively. The approximation $\tilde{L}_{i,j} \, c_{i,j}$ of the differential equation $L \, c(x_i, T_j)$ is accurate of order* 2*, i.e.*

$$L \, c(x_i, T_j) \; = \; \tilde{L}_{i,j} \, c_{i,j} \; + \; \mathcal{O}((\Delta x)^2)$$

for $i = 1, \dots, N-1$, $j = 0, \dots, M$.

Let $c_j = (c_{1,j} \, , \dots , c_{N-1,j})$ be the vector of values of the solution c of the parabolic initial–boundary–value problem (5.4.6) evaluated at the different grid points $(x_i \, , T_j)$ of the jth time step. Then, replacing $L \, c(\cdot \, , T_j)$ by its discrete approximation $\tilde{L}_j c_j$ in (5.2.12) yields a system of ordinary differential equations

$$\tfrac{\partial}{\partial T} c_j \; = \; - \left(\bar{D}_j \, c_j \; + \; \bar{B}_j \right) + \mathcal{O}((\Delta x)^2) \, , \quad j = 0, \dots, M \, .$$

An approximation of the partial derivative of c with respect to the time variable T using the forward difference quotient (5.4.7) and ignoring the terms $\mathcal{O}((\Delta x)^2)$ and $\mathcal{O}(\Delta T)$ yields the weighted difference scheme

$$\begin{aligned} \frac{z_{i,j+1} - z_{i,j}}{\Delta T} \; &= \; \tilde{L}_{i,j} \left((1-\theta) \, z_{i,j} + \theta \, z_{i,j+1} \right) \, , \\ & \qquad i = 1, \dots, N-1 \, , \; j = 0, \dots, M-1 \, , \end{aligned} \tag{5.4.26}$$

where the weighting is achieved by the parameter θ. Note that only values of θ with $0 \le \theta \le 1$ make sense. Moreover, $z_{i,j}$ denotes again the approximation of the exact solution c of (5.4.6). The scheme (5.4.26) belongs to the two level schemes since the parabolic differential equation $L \, c = c_T$ contains only first derivatives with respect to T which are approximated by using the forward difference quotient (5.4.7) [101].

Plugging the definition of $\tilde{L}_{i,j}$ in (5.4.18) into (5.4.26) yields

$$\begin{aligned}\tfrac{1}{\Delta T}\,(z_{i,j+1}-z_{i,j}) \;&=\; -\tfrac{1}{(\Delta x)^2}\,\Big(v_i^j\;\left((1-\theta)\,z_{i-1,j}+\theta\,z_{i-1,j+1}\right)+\\ &\qquad m_i^j\;\left((1-\theta)\,z_{i,j}+\theta\,z_{i,j+1}\right)+u_i^j\;\left((1-\theta)\,z_{i+1,j}+\theta\,z_{i+1,j+1}\right)\Big)\end{aligned}$$

for $i=1\,,\ldots,N-1\,,\; j=0,\ldots,M-1$. A rearrangement of the terms in the previous equation yields

$$\begin{aligned}&\theta\,\alpha\,v_i^j\,z_{i-1,j+1}+(1+\theta\,\alpha\,m_i^j)\,z_{i,j+1}+\theta\,\alpha\,u_i^j\,z_{i+1,j+1}\\ &\qquad=\;-(1-\theta)\,\alpha v_i^j\,z_{i-1,j}+(1-(1-\theta)\,\alpha\,m_i^j)\,z_{i,j}-(1-\theta)\,\alpha\,u_i^j\;z_{i+1,j}\end{aligned} \tag{5.4.27}$$

with $\alpha=\frac{\Delta T}{(\Delta x)^2}$ and $i=1\,,\ldots,N-1\,,\; j=0,\ldots,M-1$.

Using matrix notation equation (5.4.27) can be rewritten in a more compact form

$$(I+\theta\tilde{D}_j)\,z_{j+1} \;=\; (I-(1-\theta)\tilde{D}_j)\,z_j \;+\; \tilde{B}_j\,,\quad \text{for}\quad j=0,\ldots,M-1\,, \tag{5.4.28}$$

with (see (5.4.23) for the definition of D_j)

$$\tilde{D}_j \;=\; \alpha\,D_j \tag{5.4.29}$$

and $\tilde{B}_j\in I\!R^{N-1}$ containing the boundary conditions (see (5.4.25) for the definition of $z_{0,j}$ and $z_{N,j}$)

$$\begin{aligned}\tilde{B}_j \;&=\; -\alpha\,\Big(v_1^j\;\left((1-\theta)\,z_{0,j}+\theta\,z_{0,j+1}\right)\,,0,\ldots,\\ &\qquad\ldots,0,u_{N-1}^j\;\left((1-\theta)\,z_{N,j}+\theta\,z_{N,j+1}\right)\Big)^T\,.\end{aligned} \tag{5.4.30}$$

The initial vector z_0 is given by the initial condition (5.2.13) evaluated at the mesh points $(x_i\,,0)$, $i=1\,,\ldots,N-1$, i.e.

$$z_0 \;=\; c_0 \;=\; \left([1-\exp(x_1)]^+\,,\ldots,[1-\exp(x_{N-1})]^+\right)^T \tag{5.4.31}$$

where $[1-\exp(x)]^+=\max(1-\exp(x),0)$.

Lemma 5.4.2 *The finite difference scheme of the parabolic initial–boundary–value problem*

$$\begin{aligned}&\tfrac{1}{2}\,\hat{\sigma}^2(x,T)\,c_{xx}-\left(b(T)+\tfrac{1}{2}\,\hat{\sigma}^2(x,T)\right)\,c_x-d(T)\,c=c_T\,, && (x,T)\in(x_{\min}\,,x_{\max})\times(0\,,\bar{T}]\,,\\ &c(\hat{\sigma}(\cdot,\cdot);x,0) \;=\; \max(1-\exp(x),0)\,, && x\in[x_{\min}\,,x_{\max}]\\ &c(\hat{\sigma}(\cdot,\cdot);x_{\min},T) \;=\; \exp(-\int_0^T d(\tau)\,d\tau)-\\ &\qquad\qquad \exp(x_{\min})\exp(-\int_0^T r(\tau)\,d\tau)\,, && T\in[0,\bar{T}]\,,\\ &c(\hat{\sigma}(\cdot,\cdot);x_{\max},T) \;=\; 0\,, && T\in[0,\bar{T}]\,,\end{aligned}$$

using the forward difference quotient (5.4.7) *for the approximation of* c_T*, the central difference quotient* (5.4.8)*, and the symmetric central difference quotient* (5.4.9) *for approximating* c_x *and* c_{xx}*, respectively, is given by the recursive scheme*

$$(I + \theta \tilde{D}_j)\, z_{j+1} \;=\; (I - (1-\theta)\tilde{D}_j)\, z_j \;+\; \tilde{B}_j\,, \quad \textit{for} \quad j = 0, \ldots, M-1$$

with the initial element z_0 *given by the initial condition of the parabolic differential equation* (5.2.13)*, see* (5.4.31)*,* z_j*,* $\tilde{D}_j$*, and* $\tilde{B}_j$ *defined in* (5.4.21), (5.4.29)*, and* (5.4.30)*, respectively, and* $0 \le \theta \le 1$.

The scheme (5.4.28) is an explicit method only in the case $\theta = 0$, since in this case the approximations of the following time layer $j+1$ are simply obtained by matrix–vector multiplication and vector addition. For all other values of θ, i.e. $0 < \theta \le 1$, the scheme (5.4.28) represents an implicit scheme [48]. In this case the scheme (5.4.28) is well–defined if the linear systems

$$(I + \theta \tilde{D}_j)\, z_{j+1} \;=\; (I - (1-\theta)\tilde{D}_j)\, z_j \;+\; \tilde{B}_j\,, \quad j = 0\,, \ldots, M-1$$

have unique solutions z_{j+1} for $j = 0\,, \ldots, M-1$. This latter condition is equivalent to the regularity of the matrices $(I + \theta\tilde{D}_j)$, $j = 0\,, \ldots, M-1$, which is examined in Theorem 5.4.3 and Corollary 5.4.5.

Moreover, note that for $\theta = 0$ the scheme (5.4.28) represents the explicit Euler method, for $\theta = 1$ the implicit Euler method, and for $\theta = \frac{1}{2}$ the Crank–Nicolson method.

Theorem 5.4.3 *Let* Δx *and* ΔT *be the discretization size for the space and the time variable, respectively,* $\theta \in [0,1]$*, and define*

$$P := \max_{\substack{i=1\,,\ldots,N-1 \\ j=0,\ldots,M-1}} \Delta x |b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}| - \hat{\sigma}^2_{i,j} \tag{5.4.32}$$

(i) If $P \le 0$ *then* $(I + \theta\tilde{D}_j)$ *is nonsingular for* $j = 0, \ldots, M-1$.

(ii) If $P > 0$ *and* ΔT *satisfies*

$$\Delta T < \frac{(\Delta x)^2}{P}\,. \tag{5.4.33}$$

then $(I + \theta\tilde{D}_j)$ *is nonsingular for* $j = 0, \ldots, M-1$.

Remark 5.4.4 Andersen, Brotherton–Ratcliffe examine in [1] the finite difference scheme for the parabolic differential equation (5.2.1) describing the European call price function dependent on the underlying asset price and the current time. Their sufficient condition for the existence of a unique solution of the tridiagonal linear systems arising from the finite difference scheme are also given by a relationship of the discretization sizes for the time and the space domain. They derive this result by using the equivalence of the diagonal dominance and invertibility of the coefficient matrices of the tridiagonal linear systems. Moreover, for proving the diagonal dominance of the tridiagonal coefficient matrix, they show that the sum of the non–diagonal elements in one row of this matrix can be represented by $\frac{1}{2}\,(1-\theta)\,(|A+B|+|C-B|)$ with $A\,, C>0$ and B can take negative as well as positive values. Depending on the sign of B, they show that this expression takes the following form:

$$\begin{aligned}&\tfrac{1}{2}\,(1-\theta)\,(|A+B|+|C-B|)\\ &\qquad = \begin{cases}\tfrac{1}{2}\,(1-\theta)\ \max(A+C\,,2\,B-C+A)\,, & \text{if} \quad B\geq 0\\ \tfrac{1}{2}\,(1-\theta)\ \max(A+C\,,2\,|B|+C-A)\,, & \text{otherwise}\end{cases}\,.\end{aligned} \tag{5.4.34}$$

We will make use of these techniques in the proof of Theorem 5.4.3, too. However, the remaining part of the proof is different to the proof of Andersen, Brotherton–Ratcliffe in [1], since they have carried out further transformations for the coefficients appearing in the parabolic differential equation (5.2.1) by using certain features of bonds, asset forwards and European call options. In our examination, however, we will use the original continuous coefficients evaluated at the different grid points in the finite difference mesh.

Proof of Theorem 5.4.3. To verify that $(I+\theta\tilde{D}_j)$ is nonsingular for $j=0,\ldots,M-1$ it is sufficient to show that these matrices are strictly diagonally dominant, see for instance Johnson and Riess [62, Theorem 2.3]. A matrix is diagonally dominant if the absolute value of the diagonal element in each row is strictly larger than the sum of the absolute values of the non–diagonal elements. Hence, for the proof that $(I+\theta\,\tilde{D}_j)$, $j=0,\ldots,M-1$, is diagonally dominant we have to verify that for $j=0,\ldots,M-1$ the elements of $(I+\theta\,\tilde{D}_j)$ fulfil

$$\begin{aligned}\theta\,\alpha\,\left(\left|v_i^j\right|+\left|u_i^j\right|\right) &< \left|1+\theta\,\alpha\,m_i^j\right|\,, \qquad i=2\,,\ldots,N-2\,,\\ \theta\,\alpha\,\left|u_1^j\right| &< \left|1+\theta\,\alpha\,m_1^j\right|\,,\\ \theta\,\alpha\,\left|v_{N-1}^j\right| &< \left|1+\theta\,\alpha\,m_{N-1}^j\right|\,.\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\theta\,\tfrac{\alpha}{2}\,\left(\left|\hat{\sigma}_{i,j}^2+\Delta x\,(b_j+\tfrac{1}{2}\hat{\sigma}_{i,j}^2)\right|+\left|\hat{\sigma}_{i,j}^2-\Delta x\,(b_j+\tfrac{1}{2}\hat{\sigma}_{i,j}^2)\right|\right)\\ &\qquad\qquad < \left|1+\theta\,\alpha\,\left(\hat{\sigma}_{i,j}^2+(\Delta x)^2\,d_j\right)\right|\,,\end{aligned} \tag{5.4.35}$$

for $i=2\,,\ldots,N-2$ and

$$\begin{aligned}\theta\,\tfrac{\alpha}{2}\,\left|\hat{\sigma}_{1,j}^2-\Delta x\,(b_j+\tfrac{1}{2}\hat{\sigma}_{1,j}^2)\right| &< \left|1+\theta\,\alpha\,\left(\hat{\sigma}_{1,j}^2+(\Delta x)^2\,d_j\right)\right|\,,\\ \theta\,\tfrac{\alpha}{2}\,\left|\hat{\sigma}_{N-1,j}^2+\Delta x\,(b_j+\tfrac{1}{2}\hat{\sigma}_{N-1,j}^2)\right| &< \left|1+\theta\,\alpha\,\left(\hat{\sigma}_{N-1,j}^2+(\Delta x)^2\,d_j\right)\right|\,,\end{aligned} \tag{5.4.36}$$

for $j=0,\ldots,M-1$.

Let

$$\begin{aligned} F_1 &= \hat{\sigma}^2_{i,j} > 0 \\ F_2 &= \Delta x\,(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}) \;=\; \Delta x\,(r_j - d_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}) \end{aligned}$$

for an arbitrary but fixed $i \in \{2, \ldots, N-2\}$ and $j \in \{0, \ldots, M-1\}$. Then, the left hand side of (5.4.35) satisfies

$$\theta\,\tfrac{\alpha}{2}\,\left(\left|\hat{\sigma}^2_{i,j} + \Delta x\,(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right| + \left|\hat{\sigma}^2_{i,j} - \Delta x\,(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right|\right) \;=\; \theta\,\tfrac{\alpha}{2}\,\left(\,|F_1 + F_2| + |F_1 - F_2|\,\right)\,.$$

In the case $r_j + \frac{1}{2}\hat{\sigma}^2_{i,j} \geq d_j$, i.e. $F_2 \geq 0$

$$\begin{aligned} |F_1 + F_2| + |F_1 - F_2| &= F_1 + F_2 + \max\{(F_1 - F_2)\,, (F_2 - F_1)\} \\ &= \max\{2F_1\,, 2F_2\} \\ &= \max\{2F_1\,, 2|F_2|\}\,. \end{aligned}$$

If $r_j + \frac{1}{2}\hat{\sigma}^2_{i,j} < d_j$ then

$$\begin{aligned} |F_1 + F_2| + |F_1 - F_2| &= |F_1 - |F_2|\,| + |F_1 + |F_2|\,| \\ &= \max\{(F_1 - |F_2|)\,, (|F_2| - F_1)\} + F_1 + |F_2| \\ &= \max\{2F_1\,, 2|F_2|\}\,. \end{aligned}$$

Hence,

$$\theta\,\tfrac{\alpha}{2}\,\left(|F_1 + F_2| + |F_1 - F_2|\right) = \theta\,\alpha\,\max\{F_1\,, |F_2|\}\,.$$

Since we have chosen arbitrary $i \in \{2, \ldots, N-2\}$ and $j \in \{0, \ldots, M-1\}$ it follows that the left hand side of (5.4.35) satisfies

$$\begin{aligned} &\theta\,\tfrac{\alpha}{2}\,\left(\hat{\sigma}^2_{i,j} + \left|\Delta x\,(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\,\right| + \left|\hat{\sigma}^2_{i,j} - \Delta x\,(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\,\right|\right) \\ &\qquad = \theta\,\alpha\,\max\left\{\hat{\sigma}^2_{i,j}\,, \Delta x\,\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right|\right\}\,, \end{aligned} \tag{5.4.37}$$

for $i = 2, \ldots, N-2$, $j = 0, \ldots, M-1$. Note that $\Delta x > 0$. Furthermore, each of the left hand sides of the equations in (5.4.36) fulfil for $j = 0, \ldots, M-1$

$$\begin{aligned} \theta\,\tfrac{\alpha}{2}\,\left|\hat{\sigma}^2_{1,j} - \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{1,j})\,\right| &\leq \theta\,\alpha\,\max\left\{\hat{\sigma}^2_{1,j}\,, \Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{1,j}\right|\right\}\,, \\ \theta\,\tfrac{\alpha}{2}\,\left|\hat{\sigma}^2_{N-1,j} + \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{N-1,j})\,\right| &\leq \theta\,\alpha\,\max\left\{\hat{\sigma}^2_{N-1,j}\,, \Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{N-1,j}\right|\right\}\,. \end{aligned} \tag{5.4.38}$$

(i) If $P \leq 0$ then

$$\Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right| \leq \hat{\sigma}^2_{i,j}$$

for all $i = 1, \ldots, N-1\,,\; j = 0, \ldots, M-1$ and with (5.4.37) and (5.4.38) we get

$$\begin{aligned} &\theta\tfrac{\alpha}{2}\left(\left|\hat{\sigma}^2_{i,j} + \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right| + \left|\hat{\sigma}^2_{i,j} - \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right|\right) \\ &\qquad = \theta\,\alpha\,\hat{\sigma}^2_{i,j} \\ &\qquad < 1 + \theta\,\alpha\,\left(\hat{\sigma}^2_{i,j} + (\Delta x)^2\,d_j\right) \\ &\qquad = \left|1 + \theta\,\alpha\,\left(\hat{\sigma}^2_{i,j} + (\Delta x)^2\,d_j\right)\right| \end{aligned} \tag{5.4.39}$$

for $i = 1, \ldots, N-1$, $j = 0, \ldots, M-1$ so that in this case the conditions (5.4.35) and (5.4.36) are satisfied for $j = 0, \ldots, M-1$ without any restrictions concerning the discretization size ΔT of the time–space.

(ii) We now examine the case $P > 0$. Define

$$\begin{aligned} Q &= \{(i,j) \,:\, i = 1, \ldots, N-1\,,\ j = 0, \ldots, M-1\}\,, \\ Q_1 &= \left\{(i,j) \,:\, \Delta x \left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right| \ \le\ \hat{\sigma}^2_{i,j}\,,\ (i,j) \in Q\right\}\,. \end{aligned} \tag{5.4.40}$$

If $(i,j) \in Q_1$ then the verification of conditions (5.4.35) and (5.4.36), respectively, follows in the same way than in part (i) of this proof. In the case $(i,j) \in Q \backslash Q_1$, we get (note that $d_j \geq 0$)

$$\begin{aligned} &\theta\,\tfrac{\alpha}{2}\left(\left|\hat{\sigma}^2_{i,j} + \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right| + \left|\hat{\sigma}^2_{i,j} - \Delta x(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j})\right|\right) \\ &\quad= \theta\,\alpha\left(\Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right|\right) \\ &\quad= \theta\,\alpha\left(\Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right| - \hat{\sigma}^2_{i,j} + \hat{\sigma}^2_{i,j}\right) \\ &\quad= \theta\left[\tfrac{\Delta T}{(\Delta x)^2}\left(\Delta x\left|b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\right| - \hat{\sigma}^2_{i,j}\right) + \alpha\,\hat{\sigma}^2_{i,j}\right] \\ &\quad< \theta\left[\frac{\Delta x\left|b_j + \frac{1}{2}\hat{\sigma}^2_{i,j}\right| - \hat{\sigma}^2_{i,j}}{P} + \alpha\,\hat{\sigma}^2_{i,j}\right] \\ &\quad\le \theta\left(1 + \alpha\left(\hat{\sigma}^2_{i,j} + (\Delta x)^2\,d_j\right)\right) \\ &\quad\le \left|1 + \theta\,\alpha\left(\hat{\sigma}^2_{i,j} + (\Delta x)^2\,d_j\right)\right| \end{aligned}$$

such that conditions (5.4.35) and (5.4.36) also hold for $(i,j) \in Q \backslash Q_1$. Consequently, conditions (5.4.35) and (5.4.36) hold for all $(i,j) \in Q$.

Thus, we have shown that $(I + \theta\,\tilde{D}_j)$ is diagonally dominant and, hence, nonsingular for $j = 0, \ldots, M-1$. ■

Theorem 5.4.3 has the disadvantage that the upper limit (5.4.33) for the discretization size ΔT of the time space depends on the volatility parameter $\hat{\sigma}$ (see the definition of P (5.4.32)). However, the adaption of the theoretical computed European call values to the market European call values is carried out by adjusting the volatility parameter $\hat{\sigma}$. Thus, it is possible that the change of the volatility parameter during the optimization process makes a refinement of the discretization size ΔT necessary to maintain the regularity of the matrices $(I + \theta\,\tilde{D}_j)$ for $j = 0, \ldots, M-1$. Therefore, in the following corollary, we define an upper limit for ΔT which is independend of $\hat{\sigma}$. Thus, the conditions stated in Corollary 5.4.5 guarantee the regularity of $(I + \theta\,\tilde{D}_j)$, $j = 0, \ldots, M-1$ for the complete optimization process without having to change the discretization with respect to the time variable T during the optimization.

Corollary 5.4.5 *Let Δx and ΔT be the discretization size for the space and the time variable, respectively, $\theta \in [0,1]$. If $\Delta x \leq 2$ and ΔT satisfies*

$$\Delta T < \frac{\Delta x}{\|b\|_\infty} \tag{5.4.41}$$

with $\|b\|_\infty$ is the Maximum–norm in $I\!R^M$, i.e. $\|b\|_\infty = \max_{j=0,\dots,M-1} \; |b_j|$, then, $(I + \theta\, \tilde{D}_j)$ is nonsingular for $j = 0, \dots, M-1$.

Proof: It holds the following relationship between P given by (5.4.32) in Theorem 5.4.3 and $\|b\|_\infty$:

$$\begin{aligned} P &= \max_{\substack{i=1,\dots,N-1 \\ j=0,\dots,M-1}} \Delta x\, |b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}| - \hat{\sigma}^2_{i,j} \\ &= \Delta x \, \max_{i,j} \, |b_j + \tfrac{1}{2}\,\hat{\sigma}^2_{i,j}| - \frac{\hat{\sigma}^2_{i,j}}{\Delta x} \\ &\leq \Delta x \, \max_{i,j} \, |b_j| + \tfrac{1}{2}\,\hat{\sigma}^2_{i,j} - \tfrac{1}{2}\,\hat{\sigma}^2_{i,j} \\ &= \Delta x \, \|b\|_\infty \, . \end{aligned}$$

Thus, it follows

$$\Delta T \; < \; \frac{\Delta x}{\|b\|_\infty} \; \leq \; \frac{(\Delta x)^2}{P}$$

so that Corollary 5.4.5 immediately follows from Theorem 5.4.3. ■

Remark 5.4.6 To guarantee the regularity of the matrices $(I + \theta \tilde{D}_j)$, the discretization size Δx of the state variable has to be less than two. Since we have changed variables from K to x according to the transformation $K = S\exp(x)$ a discretization of the x–space with $\Delta x \leq 2$ is very reasonable to achieve a sufficient accuracy of the approximation z_j computed with the discrete scheme (5.4.28) with regard to the exact solution c_j to the parabolic differential equation (5.2.12) with initial condition (5.2.13) and the appropriate boundary conditions (5.4.5).

Moreover, the sufficient condition of the invertibility of the matrices $(I + \theta \tilde{D}_j)$, $j = 0, \dots, M-1$, with regard to the discretization size ΔT of the time variable is given by condition (5.4.41). Usually, the size of $\|b\|_\infty$ is 10^{-1} or less such that the constraint (5.4.41) is not likely to interfere with the design of the finite difference mesh.

5.4.3 Convergence of the Discretization Scheme

Using the finite difference scheme (5.4.28) to approximate the solution to the parabolic initial–boundary–value problem (5.4.6), we are now interested in the behavior of the solution z_j, $j = 0, \dots, M$, of (5.4.28), when the discretization sizes Δx and ΔT tend to zero. Hence, in this subsection, we examine the convergence behavior of the finite difference method (5.4.28) – (5.4.31), i.e. we study whether the solution z_j, $j = 0, \dots, M$, of the finite–difference scheme

(5.4.28) – (5.4.31) tends to the solution $c_j = (c_{1,j}\,, \ldots, c_{N-1,j})$, $c_{i,j} = c(x_i, T_j)$, $j = 0, \ldots, M$, of the parabolic initial–boundary–value problem (5.4.6) as we refine the discretization sizes Δx and ΔT.

Definition 5.4.7 Let c be the solution to the parabolic initial–boundary–value problem (5.4.6) and z_j, $j = 0\,, \ldots, M$, be the solution to the finite difference scheme

$$\frac{z_{i,j+1} - z_{i,j}}{\Delta T} \;=\; \tilde{L}_{i,j}\left((1-\theta)\, z_{i,j} + \theta\, z_{i,j+1}\right)\,, \quad i = 1\,, \ldots, N-1\,,\; j = 0\,, \ldots, M-1\,,$$

approximating the parabolic initial–boundary–value problem (5.4.6) with

$$z_{i,0} = \max(1 - \exp(x_i)\,, 0)\,, \quad i = 1\,, \ldots, N-1$$

and $z_{0,j}\,, z_{N,j}$, $j = 0\,, \ldots, M$ are given by (5.4.25). Then, the given finite difference method is a convergent scheme at maturity T if, as $(j+1)\,\Delta T \to T$,

$$\|z_{j+1} - c_{j+1}\| \to 0$$

as $\Delta x \to 0$ and $\Delta T \to 0$ and with $c_j = (c_{1,j}\,, \ldots, c_{N-1,j})$ and $c_{i,j} = c(x_i\,, T_j)$. Moreover, a finite difference scheme is convergent of order (p, q), $p\,, q \geq 1$ if

$$\|z_{j+1} - c_{j+1}\| \;=\; \mathcal{O}((\Delta x)^p) + \mathcal{O}((\Delta T)^q)\,.$$

Remark 5.4.8 In Definition 5.4.7, we have not specified the norm $\|\cdot\|$, yet. In the following examinations we will use the discretized L_2–norm which is given by

$$\|w\|_{2,\Delta x} \;=\; \left(\Delta x \sum_{i=1}^{N-1} w_i^2\right)^{\frac{1}{2}}\,, \tag{5.4.42}$$

for $w = (w_1, \ldots, w_{N-1})$.

Note that in Subsection 5.2.3, we have only examined the existence and uniqueness of a solution to the pure initial–value problem $L\,c = c_T$ with initial condition (5.2.13). However, by treating the differential equation numerically, we were forced to introduce boundary conditions such that we are now confronted with an initial–boundary–value problem of parabolic type. The theory of the existence and the uniqueness of solutions of these latter problems is well–established, too. They are for instance studied in the monographs by Friedman [37] or Ladyženskaja, Solonnikov and Ural'ceva [66]. In Appendix A, we briefly cite the results we will fall back on in the following examinations. To begin with, the first result in this section, i.e. Theorem A.1.1, guarantees the existence and uniqueness of a solution $c(x, T)$ to the parabolic initial–boundary–value problem (5.4.6) as well as its continuous dependence on the initial data.

Two important aspects in the examination of the convergence of the solution z_j, $j = 0, \ldots, M$, of the finite difference scheme (5.4.28) to the unique solution c_j, $j = 0, \ldots, M$, of the parabolic initial–boundary–value problem (5.4.6) are the consistency and the stability of the finite difference scheme. To begin with, we study the consistency and the stability of the finite difference scheme (5.4.28), and at the end of this subsection, we use both of these concepts to derive convergence results of the solution of the discrete scheme (5.4.28) to the solution $c(x,T)$ of the given parabolic initial–boundary–value problem.

Consistency of the Finite–Difference Scheme

A finite difference scheme is said to be a consistent scheme if the finite difference equation converges to the parabolic differential equation as the grid spacings approaches zero [31]. The examination of consistency of a finite difference scheme consists in the study of the error that occurs if the finite difference scheme is evaluated with the exact solution of the parabolic differential equation. This error is called the local truncation error of the finite difference scheme.

Definition 5.4.9 A finite difference scheme

$$\frac{z_{i,j+1} - z_{i,j}}{\Delta T} \; = \; \tilde{L}_{i,j} \left((1-\theta)\, z_{i,j} + \theta\, z_{i,j+1} \right) \; , \quad i = 1\,, \ldots , N-1\,,\; j = 0\,, \ldots , M-1\,,$$

is consistent with a parabolic initial–boundary–value problem $L\,c = c_T$ with accompanying initial and boundary conditions if for the solution $c(x,T)$ of the initial–boundary–value problem and with $c(x_i\,,T_j) = c_{i,j}$ the local truncation error $\varepsilon_{i,j+1}$ fulfils for $i = 1\,,\ldots,N-1$ and $j = 0\,,\ldots,M-1$

$$\varepsilon_{i,j+1} = \left[\frac{c_{i,j+1} - c_{i,j}}{\Delta T} - \tilde{L}_{i,j} \left((1-\theta)\, c_{i,j} + \theta\, c_{i,j+1} \right) \right] \to 0$$

as $\Delta x\,, \Delta T \to 0$. Moreover, the difference scheme is said to be accurate of order (p,q) to the given parabolic differential equation at maturity T if, as $(j+1)\,\Delta T \to T$

$$\| \varepsilon_{j+1} \| = \mathcal{O}((\Delta x)^p) + \mathcal{O}((\Delta T)^q)\,,$$

as $\Delta x \to 0$, $\Delta T \to 0$ and with ε_{j+1} denoting the vector whose ith component is $\varepsilon_{i,j+1}$.

A main tool in proving the consistency of finite difference schemes is Taylor's expansion. Applying Taylor's expansion to a function $c(x,T)$ it is necessary that $c(x,T)$ is sufficiently smooth, i.e. that partial derivatives of $c(x,T)$ exist up to a certain order and that they are continuous. According to Theorem A.1.1, we know that the solution $c(x,T)$ is a continuous function in the domain $[x_{\min}, x_{\max}] \times [0, \bar{T}]$ having two continuous x–derivatives and one continuous T–derivative in the interior $(x_{\min}, x_{\max}) \times (0, \bar{T})$ if the function describing the initial condition is continuous. Moreover, Ladyženskaja, Solonnikov, and Ural'ceva have shown in [66, IV, Theorem 5.2] (see also Appendix A) that the smoothness of the solution $c(x,T)$ to the parabolic differential equation (5.2.12) depends on the smoothness of the initial condition accompanying the differential equation, i.e. the order of differentiability of the solution $c(x,T)$ of the parabolic differential equation (5.2.12) increases with the order of differentiability of its initial condition.

Since the initial condition is evaluated at discrete points in the finite difference scheme, it is also possible that these discrete points present an approximation of a function possessing more smoothness features such as a certain order of differentiability. Hence, we will examine the consistency of the finite difference scheme (5.4.28) with respect to the parabolic differential equation (5.2.12) accompanied by an initial condition that deviates slightly from the initial condition (5.2.13) but which possesses certain smoothness characteristics. Thus, in the following examination of the consistency of the finite difference scheme (5.4.28), we will refer to the slightly perturbed parabolic initial boundary value problem

$$
\begin{aligned}
&\tfrac{1}{2}\,\hat{\sigma}^2(x,T)\,c_{xx} - \big(b(T) + \tfrac{1}{2}\,\hat{\sigma}^2(x,T)\big)\ c_x - d(T)\,c = c_T\,, \\
&\qquad\qquad\qquad\qquad\qquad\qquad (x,T) \in (x_{\min}\,, x_{\max}) \times (0\,, \bar{T}]\,, \\
c(\hat{\sigma}(\cdot,\cdot); x, 0) &= \varphi(x)\,, \qquad\qquad\qquad\qquad\qquad x \in [x_{\min}\,, x_{\max}] \\
c(\hat{\sigma}(\cdot,\cdot); x_{\min}, T) &= \exp(-\int_0^T d(\tau)\ d\tau) - \\
&\qquad \exp(x_{\min})\exp(-\int_0^T r(\tau)\ d\tau)\,, \quad T \in [0, \bar{T}]\,, \\
c(\hat{\sigma}(\cdot,\cdot); x_{\max}, T) &= 0\,, \qquad\qquad\qquad\qquad\qquad T \in [0, \bar{T}]\,,
\end{aligned}
\tag{5.4.43}
$$

with $\varphi(x)$ sufficiently smooth and

$$\max_{x \in [x_{\min}\,, x_{\max}]} |\max(1 - \exp(x), 0) - \varphi(x)| \leq \delta\,.$$

Let $x_- = \max(x_i\,,\ i = 0, \ldots, N\ :\ x_i < 0)$ and $x_+ = \min(x_i\,,\ i = 0, \ldots, N\ :\ x_i \geq 0)$, then $\varphi(x)$ is chosen such that it agrees with $\max(1 - \exp(x), 0)$ on $[x_{\min}\,, x_-] \cup [x_+\,, x_{\max}]$ and in the domain (x_-, x_+), it approximates $\max(1 - \exp(x), 0)$ by satisfying certain smoothness conditions such as the existence of continuous partial derivatives of a certain order on $[x_{\min}\,, x_{\max}]$.

Let $c(x,T)$ be the solution to the original parabolic initial–boundary–value problem (5.4.6) and $\tilde{c}(x,T)$ be the solution to the perturbed initial–boundary–value problem (5.4.43). Then, according to Theorem A.1.1, we know that the difference between $c(x,T)$ and $\tilde{c}(x,T)$ on the complete domain $[x_{\min}\,, x_{\max}] \times [0\,, \bar{T}]$ is bounded above by the maximal difference between the original initial condition $c(x,0) = \max(1 - \exp(x), 0)$ and its smooth approximation $\varphi(x)$, i.e.

$$\max_{(x,T) \in [x_{\min}\,, x_{\max}] \times [0\,, \bar{T}]} |c(x,T) - \tilde{c}(x,T)|\ \leq\ \delta\,.$$

Now, in the next theorem, we examine the consistency of the finite difference scheme (5.4.28) – (5.4.31) with respect to the perturbed initial–boundary–value problem (5.4.43).

Theorem 5.4.10 *Let the initial condition* $\varphi(x)$ *of* (5.4.43) *satisfy* $\varphi \in H^{4+\epsilon}([x_{\min}\,, x_{\max}])$*, with* $\epsilon > 0$ *arbitrary small. Then, the finite difference scheme* (5.4.28) – (5.4.31) *is consistent to the* *perturbed initial–boundary–value problem* (5.4.43) *of order* $(2,1)$ *with respect to* $\|\cdot\|_{2,\Delta x}$ *for* $0 \leq \theta \leq 1$.

Moreover, if $\varphi \in H^{6+\epsilon}([x_{\min}\,, x_{\max}])$, $\epsilon > 0$ *arbitrary small, and the coefficients* $b(T)$, $\hat{\sigma}(x,T)$ *and* $d(T)$ *in the parabolic differential equation* (5.2.12) *are not dependent on* T, *i.e.*

$$b(T) = \bar{b}\,, \quad \hat{\sigma}(x,T) = \bar{\sigma}(x)\,, \quad d(T) = \bar{d}$$

then the finite difference scheme (5.4.28) – (5.4.31) *is consistent to the perturbed initial–boundary–value problem* (5.4.43) *of order* $(2,2)$ *with respect to* $\|\cdot\|_{2,\Delta x}$ *for* $\theta = \frac{1}{2}$, *i.e. in this case the Crank–Nicolson method is consistent of order* $(2,2)$ *with respect to* $\|\cdot\|_{2,\Delta x}$.

Remark 5.4.11 The spaces $H^l([x_{\min}\,, x_{\max}])$ are Banach spaces whose elements are functions that are continuous in the sense of Hölder in $(x_{\min}\,, x_{\max})$, which have in $[x_{\min}\,, x_{\max}]$ continuous derivatives up to order $[l]$ $(= \max\{z \in \mathbb{Z} : z \leq l\})$ inclusively and which are finite with respect to a certain norm defined on $H^l([x_{\min}\,, x_{\max}])$, see Appendix A for a definition of this norm.

We have dispersed the proof of Theorem 5.4.10 into Appendix B since we have used the usual proving techniques applied to the determination of the consistency of finite difference schemes and its order of accuracy.

However, note that in the proof of Theorem 5.4.10, it is necessary that the solution of the examined initial–boundary–value problem has to be differentiable at $T = 0$. This condition is not satisfied for the original parabolic initial–boundary–value problem (5.4.6), since its initial condition $c(x,0) = \max(1 - \exp(x), 0)$ is continuous on $[x_{\min}\,, x_{\max}]$ but not differentiable at $x = 0$. In this case, the local truncation error for the first time layer can not be determined by using Taylor's expansion for an evaluation of the accuracy of the finite difference quotients approximating the different differentiation operators appearing in the parabolic differential equation (5.2.12). This aspect is often not considered in the examination of finite difference schemes that are applied to parabolic differential equations with a nonsmooth initial condition, see for instance Wilmott et al. [107], Andersen, Brotherton–Ratcliffe [1] for an application of finite difference schemes in option pricing models. In this case, the well–known order of accuracy of $(2,1)$ of the finite difference schemes (5.4.28) in general and the order $(2,2)$ in the special case of $\theta = \frac{1}{2}$ cannot be maintained when approaching $T = 0$. We will return to this aspect at the end of this section.

Stability of the Discretization Scheme

Next, we examine the stability of the finite difference scheme (5.4.28)

$$(I + \theta \tilde{D}_j)\, z_{j+1} \;=\; (I - (1-\theta)\tilde{D}_j)\, z_j \;+\; \tilde{B}_j\,, \quad j = 0\,, \ldots, M-1\,,$$

with $\tilde{D}_j$, $\tilde{B}_j$, and z_j defined in (5.4.29), (5.4.30), and (5.4.21), respectively. In Corollary 5.4.5 we have shown that the matrices $(I + \theta \tilde{D}_j)$, $j = 0\,, \ldots, M-1$, are nonsingular if $\Delta T < \Delta x / \|b\|_\infty$. If this condition is satisfied, formally the finite difference scheme (5.4.28) can be rewritten as

$$z_{j+1} \;=\; (I + \theta \tilde{D}_j)^{-1}\, (I - (1-\theta)\tilde{D}_j)\, z_j \;+\; (I + \theta \tilde{D}_j)^{-1}\, \tilde{B}_j\,, \quad j = 0\,, \ldots, M-1$$

using the inverse of $(I + \theta \tilde{D}_j)$, $j = 0, \ldots, M-1$.

A discretization scheme is said to be stable if small errors in the initial conditions cause small errors in the solution [101]. Hence, the finite difference scheme (5.4.28) is stable if the absolute values of the eigenvalues of the matrices

$$\begin{aligned} \mathcal{D}_j &:= (I + \theta \tilde{D}_j)^{-1} \, (I - (1-\theta)\tilde{D}_j) \\ &= (I + \theta\alpha D_j)^{-1} \, (I - (1-\theta)\alpha D_j) \end{aligned} \tag{5.4.44}$$

$j = 0, \ldots, M-1$, are less than one [94], [103], i.e. let λ_ν^j, $\nu = 1, \ldots, N-1$, be the eigenvalues of $\mathcal{D}_j$, $j = 0, \ldots, M-1$, then the finite difference scheme is stable if

$$|\lambda_\nu^j| < 1 \quad \text{for} \quad \nu = 1, \ldots, N-1, \; j = 0, \ldots, M-1 \,. \tag{5.4.45}$$

If μ_ν^j are the eigenvalues of D_j, $j = 0, \ldots, M-1$, then the eigenvalues λ_ν^j of $\mathcal{D}_j$, $j = 0, \ldots, M-1$, are given by

$$\lambda_\nu^j = \frac{1 - (1-\theta)\,\alpha\,\mu_\nu^j}{1 + \theta\,\alpha\,\mu_\nu^j}\,, \quad \nu = 1, \ldots, N-1\,.$$

We start the investigation of the stability of the finite difference system (5.4.28) by examining the eigenvalues μ_ν^j, $\nu = 1, \ldots, N-1$, of D_j, $j = 0, \ldots, M-1$.

Lemma 5.4.12 *Let Δx and ΔT be the discretization size for the space and the time variable, respectively, and $\theta \in [0, 1]$. If*

$$\Delta x < \min_{\substack{i=1,\ldots,N-1 \\ j=0,\ldots,M-1}} \frac{\hat{\sigma}_{i,j}^2}{|b_j + \frac{1}{2}\,\hat{\sigma}_{i,j}^2|}\,, \tag{5.4.46}$$

then, the matrices D_j, $j = 0, \ldots, M-1$, are positive semidefinite. Moreover, the eigenvalues μ_ν^j, $\nu = 1, \ldots, N-1$, of D_j, $j = 0, \ldots, M-1$, satisfy

$$0 \le d_j\,(\Delta x)^2 \le \mu_\nu^j \le \max_{i=1,\ldots,N-1} 2\,\hat{\sigma}_{i,j}^2 + (\Delta x)^2\, d_j = 2\,\|\hat{\sigma}_j^2\|_\infty + (\Delta x)^2\, d_j \tag{5.4.47}$$

with $\|\hat{\sigma}_j^2\|_\infty = \max_{i=1,\ldots,N-1} \hat{\sigma}_{i,j}^2$.

Proof: Firstly, we show that the matrices D_j, $j = 0, \ldots, M-1$, are similar to a symmetric matrix. For this purpose, we introduce the diagonal matrices

$$Y_j = \mathrm{diag}(y_1^j, \ldots, y_{N-1}^j)\,, \quad j = 0, \ldots, M-1$$

where the diagonal elements y_i^j, $i = 1, \ldots, N-1$, $j = 0, \ldots, M-1$, are recursively defined by

$$\begin{aligned} y_1^j &= 1\,, & j &= 0, \ldots, M-1\,, \\ y_i^j &= y_{i-1}^j \sqrt{\frac{v_i^j}{u_{i-1}^j}}\,, & i &= 2, \ldots, N-1\,,\; j = 0, \ldots, M-1\,. \end{aligned}$$

The diagonal elements y_i^j, $i = 1, \dots, N-1$, of Y_j, $j = 0\,, \dots, M-1$, are well–defined since the condition (5.4.46) on the discretization size Δx guarantees that

$$\begin{aligned} u_i^j &= -\tfrac{1}{2}\left(\hat\sigma_{i,j}^2 - \Delta x\left(b_j + \tfrac{1}{2}\hat\sigma_{i,j}^2\right)\right) < 0 \\ v_i^j &= -\tfrac{1}{2}\left(\hat\sigma_{i,j}^2 + \Delta x\left(b_j + \tfrac{1}{2}\hat\sigma_{i,j}^2\right)\right) < 0 \end{aligned}$$

for all $i = 1\,, \dots, N-1\,,\ j = 0\,, \dots, M-1$ regardless if $(b_j + \frac{1}{2}\hat\sigma_{i,j}^2) > 0$ or $(b_j + \frac{1}{2}\hat\sigma_{i,j}^2) < 0$ such that $\frac{v_i^j}{u_{i-1}^j} > 0$ for all $i = 1\,, \dots, N-1\,,\ j = 0\,, \dots, M-1$. Then,

$$Y_j^{-1}\, D_j\, Y_j = \begin{pmatrix} m_1^j & \tilde u_1^j & & & \\ \tilde v_2^j & m_2^j & \tilde u_2^j & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde v_{N-2}^j & m_{N-2}^j & \tilde u_{N-2}^j \\ & & & \tilde v_{N-1}^j & m_{N-1}^j \end{pmatrix}$$

with

$$\tilde v_{i+1}^j = \frac{y_i^j}{y_{i+1}^j}\, v_{i+1}^j = \sqrt{\frac{u_i^j}{v_{i+1}^j}}\, v_{i+1}^j = \frac{u_i^j}{v_{i+1}^j}\sqrt{\frac{v_{i+1}^j}{u_i^j}}\, v_{i+1}^j = \frac{y_{i+1}^j}{y_i^j}\, u_i^j = \tilde u_i^j$$

for $i = 1\,, \dots, N-1\,,\ j = 0\,, \dots, M-1$. Thus, D_j is similar to a symmetric matrix. Since similar matrices have identical characteristic polynomials and the eigenvalues of symmetric matrices are real, it follows that the eigenvalues of D_j, $j = 0, \dots, M-1$, are real.

Moreover, the Gershgorin Circle Theorem, see for instance Golub, van Loan [43, Theorem 7.2.1], applied to D_j, $j = 0\,, \dots, M-1$, yields that the eigenvalues μ_ν^j, $\nu = 1\,, \dots, N-1$, of D_j, $j = 0\,, \dots, M-1$, satisfy

$$\mu_\nu^j \in \bigcup_{i=1}^{N-1} \mathcal{G}_i^j$$

with the Gershgorin circles $\mathcal{G}_i^j$ given by

$$\begin{aligned} \mathcal{G}_i^j &= \left\{\mu \,:\, |\mu - m_i^j| \le |v_i^j| + |u_i^j|\right\}\,, \quad i = 2\,, \dots, N-2\,, \\ \mathcal{G}_1^j &= \left\{\mu \,:\, |\mu - m_1^j| \le |u_1^j|\right\}\,, \\ \mathcal{G}_{N-1}^j &= \left\{\mu \,:\, |\mu - m_{N-1}^j| \le |v_{N-1}^j|\right\}\,, \end{aligned}$$

for $j = 0\,, \dots, M-1$. Thus, for $\mu \in \mathcal{G}_i^j$ holds

$$\begin{aligned} \left|\mu - \hat\sigma_{i,j}^2 - (\Delta x)^2\, d_j\right| &\le \left|\tfrac{1}{2}\left(\hat\sigma_{i,j}^2 + \Delta x\,(b_j + \tfrac{1}{2}\hat\sigma_{i,j}^2)\right)\right| + \left|\tfrac{1}{2}\left(\hat\sigma_{i,j}^2 - \Delta x\,(b_j + \tfrac{1}{2}\hat\sigma_{i,j}^2)\right)\right| \\ &= \hat\sigma_{i,j}^2 \end{aligned}$$

since $\frac{1}{2}\left(\hat\sigma^2_{i,j} + \Delta x\,(b_j + \frac{1}{2}\hat\sigma^2_{i,j})\right) > 0$ and $\frac{1}{2}\left(\hat\sigma^2_{i,j} - \Delta x\,(b_j + \frac{1}{2}\hat\sigma^2_{i,j})\right) > 0$. Thus,

$$(\Delta x)^2\, d_j \;\le\; \mu \;\le\; 2\,\hat\sigma^2_{i,j} + (\Delta x)^2\, d_j \quad \text{for} \quad \mu \in \mathcal{G}^j_i$$

such that

$$\bigcup_{i=1}^{N-1} \mathcal{G}^j_i \;\subseteq\; \left[(\Delta x)^2\, d_j\,,\, 2\,\|\hat\sigma^2_j\|_\infty + (\Delta x)^2\, d_j\right]$$

which proves that the matrices D_j, $j = 0, \dots, M-1$ are positive semidefinite, and that condition (5.4.47) is satisfied for the eigenvalues μ^j_ν, $\nu = 1\,, \dots, N-1$, of D_j, $j = 0\,, \dots, M-1$. ■

Remark 5.4.13 If $\hat\sigma^2_{i,j} \ge |b_j|$ then $\frac{|b_j|}{\hat\sigma^2_{i,j}} \le 1$ such that

$$\frac{\hat\sigma^2_{i,j}}{|b_j + \frac{1}{2}\,\hat\sigma^2_{i,j}|} \;=\; \frac{1}{\left|\frac{b_j}{\hat\sigma^2_{i,j}} + \frac{1}{2}\right|} \;\ge\; \frac{1}{\frac{|b_j|}{\hat\sigma^2_{i,j}} + \frac{1}{2}} \;\ge\; \frac{1}{\frac{3}{2}} \;=\; \frac{2}{3}\,.$$

Thus, in this case, condition (5.4.46) is satisfied if the discretization size Δx of the state variable fulfils $\Delta x < \frac{2}{3}$. However, this is only a weak restriction on the discretization size Δx since we have changed variables from K to x according to the transformation $K = S\,\exp(x)$, see also Remark 5.4.6.

Condition (5.4.46) becomes more restrictive for the discretization size Δx if $\hat\sigma^2_{i,j} < |b_j|$. In this case, the size of the quotient $\frac{|b_j|}{\hat\sigma^2_{i,j}}$ determines the size of Δx. If $\frac{|b_j|}{\hat\sigma^2_{i,j}} \approx 10$ then condition (5.4.46) is satisfied if $\Delta x < 0.09$. However, the numerical results in Section 5.6 will show that with a reasonable chosen number of discretization points this condition will not be neglected, neither.

Having information available about the eigenvalues of the matrices D_j, $j = 0\,, \dots, M-1$, now we can examine the eigenvalues of the matrices $\mathcal{D}_j$, $j = 0\,, \dots, M-1$, and, consequently, the stability of the finite difference scheme (5.4.28). The conditions ensuring the stability for the different finite difference schemes, i.e. for $0 \le \theta \le 1$, are stated in the following theorem.

Theorem 5.4.14 *Let the assumptions of Lemma 5.4.12 be satisfied and let μ^j_ν, $\nu = 1\,, \dots, N-1$, be the eigenvalues of the matrices D_j, $j = 0, \dots, M-1$. Then, the finite difference scheme* (5.4.28) – (5.4.31) *is absolutely stable for $\frac{1}{2} \le \theta \le 1$ and it is stable for $0 \le \theta < \frac{1}{2}$ if the discretization size ΔT of the time step satisfies*

$$\Delta T \;<\; \frac{2\,(\Delta x)^2}{(1 - 2\,\theta)\,(2\,\|\hat\sigma^2\|_\infty + \|d\|_\infty (\Delta x)^2)} \tag{5.4.48}$$

with

$$\|\hat\sigma^2\|_\infty = \max_{\substack{i=1\,,\dots,N-1\\ j=0,\dots,M-1}} \hat\sigma^2_{i,j} \quad \textit{and} \quad \|d\|_\infty = \max_{j=0,\dots,M-1} |d_j|\,.$$

Proof: We have already mentioned above that the finite difference scheme (5.4.28) – (5.4.31) is stable if the eigenvalues λ_ν^j of $\mathcal{D}_j$ (see (5.4.44)) are less than one. If μ_ν^j are the eigenvalues of D_j, $j = 0\,,\ldots,M-1$, then the stability condition for the finite difference scheme (5.4.28) – (5.4.31) is given by

$$\left|\lambda_\nu^j\right| \;=\; \left|\frac{1-(1-\theta)\,\alpha\,\mu_\nu^j}{1+\theta\,\alpha\,\mu_\nu^j}\right| < 1\,, \quad \text{for} \quad \nu = 1\,,\ldots,N-1\,, \quad j = 0\,,\ldots,M-1\,.$$

This condition is satisfied if and only if

$$\left|1-(1-\theta)\,\alpha\,\mu_\nu^j\right| \;<\; \left|1+\theta\,\alpha\,\mu_\nu^j\right| \;=\; 1+\theta\,\alpha\,\mu_\nu^j$$

respective

$$-1-\theta\,\alpha\,\mu_\nu^j \;<\; 1-(1-\theta)\,\alpha\,\mu_\nu^j \;<\; 1+\theta\,\alpha\,\mu_\nu^j\,,$$

which is equivalent to

$$(1-2\,\theta)\;\mu_\nu^j \;<\; \frac{2}{\alpha}\,.$$

This inequality is satisfied for all $\frac{1}{2} \le \theta \le 1$ without any further restriction with regard to the discretization sizes ΔT and Δx since $\alpha = \frac{\Delta T}{(\Delta x)^2} > 0$, $\mu_\nu^j \ge 0$ for $\nu = 1,\ldots,N-1$, $j = 0,\ldots,M-1$ according to Lemma 5.4.12 and $1-2\,\theta < 0$ for $\theta \ge \frac{1}{2}$. For $0 \le \theta < \frac{1}{2}$ the stability condition is satisfied if

$$\frac{2}{(1-2\,\theta)\,\alpha} \;>\; 2\,\|\hat{\sigma}^2\|_\infty + \|d\|_\infty(\Delta x)^2 \;\ge\; \max_{\substack{\nu=1\,,\ldots,N-1\\ j=0,\ldots,M-1}} \mu_\nu^j$$

which is equivalent to

$$\frac{2\,(\Delta x)^2}{(1-2\,\theta)\,(2\,\|\hat{\sigma}^2\|_\infty + \|d\|_\infty(\Delta x)^2)} \;>\; \Delta T\,.$$

Thus, Theorem 5.4.14 is proven. ■

Remark 5.4.15 The stability condition (5.4.48) puts for $0 \le \theta < \frac{1}{2}$ strong restrictions on the size of the discretization size ΔT since this condition means that maintaining stability requires to quarter the size of ΔT if only the number of discretization points in the x–mesh are doubled. Regarding computational effort, this means that on the one side the computation of z_j for one time–step requires twice as much effort (assuming that the computational effort for the solution of the tridiagonal system is $\mathcal{O}(N)$) and on the other side the computation of four times as many time–steps. Thus, maintaining stability in this case means that the computational effort is eight times as much than it was before if we started from a stable system and doubled the number of discretization points in the x–mesh. Hence, a choice of θ with $0 < \theta < \frac{1}{2}$ is not recommendable since on the one side in each time–step a tridiagonal system has to be solved and on the other side obtaining a higher accuracy by increasing the number of discretization points in the x–mesh results in a considerable increase of the computational effort. The increase of the computational cost might be less when $\frac{1}{2} \le \theta \le 1$ since then the resulting finite difference scheme is unconditional stable and does not require urgently an adaption of the discretization size of ΔT when the discretization size Δx is altered.

Remark 5.4.16 If we would have applied a finite difference method to the original parabolic differential equation (5.2.7)

$$C_T - \tfrac{1}{2}\sigma(K,T)^2 K^2 C_{KK} + (r(T) - d(T))KC_K + d(T)C = 0$$

then, by using the forward difference quotients, the central difference quotient and the symmetric central difference quotient for the approximation of C_T, C_K and C_{KK}, respectively, and discretizing the K–space by

$$K_i = i\,\Delta K\,, \quad \text{for } i = 0\,,\ldots,N \quad \text{and with } K_{\max} = N\,\Delta K$$

the finite difference scheme would be given by

$$\begin{aligned} &\theta\,\Delta T\,\hat{v}_i^j\,z_{i-1,j+1} + (1+\theta\,\Delta T\,\hat{m}_i^j)\,z_{i,j+1} + \theta\,\Delta T\,\hat{u}_i^j\,z_{i+1,j+1} \\ &\qquad = \; -(1-\theta)\,\Delta T\hat{v}_i^j\,z_{i-1,j} + (1-(1-\theta)\,\Delta T\,\hat{m}_i^j)\,z_{i,j} - (1-\theta)\,\Delta T\,\hat{u}_i^j\,z_{i+1,j} \end{aligned}$$

with, $i = 1\,,\ldots,N-1$, $j = 0\,,\ldots,M-1$,

$$\begin{aligned} \hat{v}_i^j &= -\tfrac{1}{2}\left(\hat{\sigma}_{i,j}^2\,i^2 + b_j\,i\right)\,, \\ \hat{m}_i^j &= \hat{\sigma}_{i,j}^2\,i^2 + d_j\,, \\ \hat{u}_i^j &= -\tfrac{1}{2}\left(\hat{\sigma}_{i,j}^2\,i^2 - b_j\,i\right)\,. \end{aligned}$$

In this case, condition (5.4.46) in Lemma 5.4.12 would become $i > \max(|b_j|/\hat{\sigma}_{i,j}^2)$ which puts restrictions on the choice of the lower bound of the domain $[K_{\min}, K_{\max}]$. This, however, is very much undesirable.

Moreover, the condition (5.4.48) on the discretization size ΔT in Theorem 5.4.14 would become for $0 \le \theta < \frac{1}{2}$

$$\Delta T \;<\; \frac{2}{(1-2\,\theta)\,(2\,\|\hat{\sigma}^2\|_\infty\,(N-1)^2 + \|d\|_\infty)} \tag{5.4.49}$$

which would mean that the maximum stable discretization size ΔT now depends on the number of discretization points in the K–space and not on the discretization size ΔK as in Theorem 5.4.14. However, we will see in the next paragraph that the convergence of a finite difference method depends on the size of ΔK such that the convergence and the stability criterion are distinct [107]. Furthermore, condition (5.4.49) makes clear that it is possible that the difference scheme is stable for small K and unstable for large K, i.e. if $\Delta T = 2/((1-2\,\theta)\,(2\,\|\hat{\sigma}^2\|_\infty\,(\frac{N}{2})^2 + \|d\|_\infty))$ the finite difference scheme would be stable for K_i, $i = 0\,,\ldots,\frac{N}{2}$ and unstable for K_i, $i = \frac{N}{2}+1,\ldots,N-1$ such that at first sight it is not clear whether the scheme is stable.

All these difficulties do not appear when applying finite difference method to the transformed version (5.2.12) of the parabolic differential equation (5.2.7) which elucidates the necessity of the change of variables.

Convergence of the Finite–Difference Scheme

After establishing the consistency and the stability of the finite difference scheme (5.4.28) – (5.4.31) we now derive the convergence of this scheme.

Theorem 5.4.17 *Let $c(x,T)$ be the solution to the parabolic initial–boundary–value problem* (5.4.6) *and let $z_j = (z_{1,j}\,,\dots,z_{N-1,j})$ be the solution to the finite difference scheme* (5.4.28) – (5.4.31)*. Moreover, assume there exists a function $\varphi \in H^{4+\epsilon}([x_{\min}\,,x_{\max}])$, $\epsilon > 0$ arbitrary small, with $\varphi(x_i) = \max(1-\exp(x_i)\,,0)$, $i = 0\,,\dots,N$, and*

$$\max_{x\in[x_{\min}\,,x_{\max}]} |\max(1-\exp(x),0)-\varphi(x)| \;\le\; (\Delta x)^s\,, \quad \textit{for some}\;\; s>0\,.$$

Suppose that the discretization sizes for the time domain ΔT and for the space domain Δx fulfil

$$\Delta x \;<\; \min_{\substack{i=1\,,\dots,N-1\\ j=0\,,\dots,M-1}} \frac{\hat\sigma^2_{i,j}}{|b_j + \frac{1}{2}\,\hat\sigma^2_{i,j}|}$$

and

$$\Delta T \;<\; \begin{cases} \min\left(\dfrac{\Delta x}{\|b\|_\infty}\,,\dfrac{2\,(\Delta x)^2}{(1-2\,\theta)\,(2\,\|\hat\sigma^2\|_\infty + \|d\|_\infty(\Delta x)^2)}\right)\,, & \textit{if}\;\; 0\le\theta<\frac{1}{2}\\ \dfrac{\Delta x}{\|b\|_\infty}\,, & \textit{if}\;\; \frac{1}{2}\le\theta\le 1\end{cases}$$

Then, the finite difference method (5.4.28) – (5.4.31) *is a convergent scheme of order* $(\min\{s\,,2\}\,,1)$ *with respect to the discretized L_2–norm $\|\cdot\|_{2,\Delta x}$, i.e.*

$$\|z_{j+1}-c_{j+1}\|_{2,\Delta x} = \mathcal{O}((\Delta x)^{\min\{s\,,2\}}) + \mathcal{O}(\Delta T)\,, \quad \textit{for} \quad j=0,\dots,M-1\,.$$

Proof: Let $c(x,T)$ be the solution to the parabolic initial–boundary–value problem (5.4.6), and let $\tilde c(x,T)$ be the solution to the slightly perturbed parabolic initial–boundary–value problem (5.4.43) with initial condition $\tilde c(x,0)=\varphi(x)$. Furthermore, let

$$z_j = (z_{1,j}\,,\dots,z_{N-1,j})$$

be the solution to the finite difference scheme

$$(I+\theta\tilde D_j)\;z_{j+1} \;=\; (I-(1-\theta)\tilde D_j)\;z_j \;+\; \tilde B_j\,, \quad j=0,\dots,M-1$$

with z_0 given by the initial condition (5.2.13)

$$z_0 \;=\; c_0 \;=\; \left([1-\exp(x_1)]^+\,,\dots,[1-\exp(x_{N-1})]^+\right)^T\,,$$

$[1-\exp(x)]^+ = \max(1-\exp(x),0)$, and $\tilde D_j$ and $\tilde B_j$ defined by (5.4.29) and (5.4.30), respectively.

If ε_{j+1} is the vector containing the local truncation errors $\varepsilon_{i,j+1}$ of the finite difference scheme (5.4.28) – (5.4.31) with respect to the slightly perturbed parabolic initial–boundary–value problem (5.4.43) in the $(j+1)$st time layer, then for the solution $\tilde{c}(x,T)$ of (5.4.43) holds

$$(I+\theta\tilde{D}_j)\,\tilde{c}_{j+1} \;=\; (I-(1-\theta)\tilde{D}_j)\,\tilde{c}_j \;+\; \tilde{B}_j \;+\; \Delta T\,\varepsilon_{j+1}\,, \quad j=0,\ldots,M-1$$

with $\varepsilon_{j+1} = (\varepsilon_{1,j+1}\,,\ldots,\varepsilon_{N-1,j+1})$ and $\tilde{c}_{j+1} = (\tilde{c}_{1,j+1}\,,\ldots,\tilde{c}_{N-1,j+1})$, $\tilde{c}_{i,j} = \tilde{c}(x_i\,,T_j)$, i.e. $\tilde{c}(x,T)$ fulfils the finite difference scheme (5.4.28) – (5.4.31) extended with a term containing the local truncation errors.

Moreover, since $\Delta x < \min_{i,j} \frac{\hat{\sigma}^2_{i,j}}{|b_j+\frac{1}{2}\,\hat{\sigma}^2_{i,j}|}$ according to Theorem 5.4.14 we have

$$\|\mathcal{D}_j\|_2 \;=\; \|(I+\theta\tilde{D}_j)^{-1}\,(I-(1-\theta)\tilde{D}_j)\|_2 \;<\; 1\,, \quad \text{for} \quad j=0,\ldots,M-1\,.$$

Note, that the matrix norms $\|\cdot\|_2$ and $\|\cdot\|_{2,\Delta x}$ (i.e. these are the matrix norms defined in terms of the vector norms) are equal since ($\|\cdot\|_{2,\Delta x} = \sqrt{\Delta x}\,\|\cdot\|_2$)

$$\begin{aligned}
\|\mathcal{D}_j\|_{2,\Delta x} &= \max_{\|y\|_{2,\Delta x}\le 1} \|\mathcal{D}_j\,y\|_{2,\Delta x} \\
&= \max_{\sqrt{\Delta x}\,\|y\|_2\le 1} \sqrt{\Delta x}\,\|\mathcal{D}_j\,y\|_2 \\
&= \max_{\|y\|_2\le 1} \|\mathcal{D}_j\,y\|_2 \\
&= \|\mathcal{D}_j\|_2
\end{aligned}$$

Now, let $e_{i,j}$ be the difference of the exact solution $\tilde{c}_{i,j}$ of the perturbed parabolic initial–boundary value problem (5.4.43) and the solution $z_{i,j}$ of the finite difference method (5.4.28) – (5.4.31), i.e.

$$e_{i,j} \;=\; \tilde{c}_{i,j} \;-\; z_{i,j}\,,$$

then $e_{i,j}$ satisfies the nonhomogeneous finite difference equation ($e_j = (e_{1,j}\,,\ldots,e_{N-1,j})$)

$$(I+\theta\tilde{D}_j)\,e_{j+1} \;=\; (I-(1-\theta)\tilde{D}_j)\,e_j \;+\; \Delta T\,\varepsilon_{j+1}\,, \quad j=0,\ldots,M-1$$

with homogeneous initial and boundary conditions, i.e. $e_0 = 0$. Thus,

$$e_{j+1} \;=\; (I+\theta\tilde{D}_j)^{-1}\,(I-(1-\theta)\tilde{D}_j)\,e_j \;+\; \Delta T\,(I+\theta\tilde{D}_j)^{-1}\,\varepsilon_{j+1}\,, \quad j=0,\ldots,M-1\,,$$

and taking the discretized L_2–norm $\|\cdot\|_{2,\Delta x}$ of e_{j+1} yields for $j=0,\ldots,M-1$

$$\begin{aligned}
\|e_{j+1}\|_{2,\Delta x} &\le \|(I+\theta\tilde{D}_j)^{-1}\,(I-(1-\theta)\tilde{D}_j)\|_{2,\Delta x}\,\|e_j\|_{2,\Delta x} \\
&\qquad + \;\Delta T\,\|(I+\theta\tilde{D}_j)^{-1}\|_{2,\Delta x}\,\|\varepsilon_{j+1}\|_{2,\Delta x} \qquad (5.4.50) \\
&\le k_1\,\|e_j\|_{2,\Delta x} + \Delta T\,\|\varepsilon_{j+1}\|_{2,\Delta x}
\end{aligned}$$

with $k_1 < 1$. To see that $\|(I+\theta\tilde{D}_j)^{-1}\|_{2,\Delta x} = \|(I+\theta\tilde{D}_j)^{-1}\|_2 \le 1$ we examine the eigenvalues λ^j_ν, $\nu = 1\,,\ldots,N-1$, of $(I+\theta\tilde{D}_j)^{-1} = (I+\theta\,\alpha\,D_j)^{-1}$, $j = 0\,,\ldots,M-1$. Let μ^j_ν,

$\nu = 1\,,\dots,N-1$, be the eigenvalues of D_j (see (5.4.23) for the definition of D_j). Lemma 5.4.12 yields that

$$0 \;\le\; \mu^j_\nu \;\le\; 2\,\|\hat{\sigma}^2_j\|_\infty + (\Delta x)^2\, d_j\,, \quad \nu = 1\,,\dots,N-1\,,$$

and, thus, for the eigenvalues of $(I + \theta\,\alpha\,D_j)^{-1}$, $j = 0\,,\dots,M-1$, holds

$$\lambda^j_\nu \;=\; \frac{1}{1+\theta\,\alpha\,\mu^j_\nu} \;\le\; 1\,.$$

Now, applying (5.4.50) recursively to $\|e_l\|_{2,\Delta x}$, $l = j,\dots,1$, yields (note $M = \frac{\bar{T}}{\Delta T}$)

$$\begin{aligned}
\|e_{j+1}\|_{2,\Delta x} &\le k_1\,\|e_j\|_{2,\Delta x} + \Delta T\,\|\varepsilon_{j+1}\|_{2,\Delta x} \\
&\le k_1\,(k_1\,\|e_{j-1}\|_{2,\Delta x} + \Delta T\,\|\varepsilon_j\|_{2,\Delta x}) + \Delta T\,\|\varepsilon_{j+1}\|_{2,\Delta x} \\
&\le \Delta T\,\sum_{l=1}^{j+1} k_1^{j-l+1}\,\|\varepsilon_l\|_{2,\Delta x} \\
&\le \Delta T\,\sum_{l=1}^{j+1}\|\varepsilon_l\|_{2,\Delta x}
\end{aligned}$$

According to Theorem 5.4.10, the accuracy of the finite difference scheme (5.4.28) – (5.4.31) with respect to the slightly perturbed initial–boundary–value problem (5.4.43) is of order $(2,1)$, i.e. $\|\varepsilon_l\|_{2,\Delta x} = \mathcal{O}((\Delta x)^2) + \mathcal{O}(\Delta T)$, $l = 1,\dots,M$, such that

$$\begin{aligned}
\|e_{j+1}\|_{2,\Delta x} &\le \Delta T\,\sum_{l=1}^{j+1}\mathcal{O}((\Delta x)^2) + \mathcal{O}(\Delta T) \\
&\le \Delta T\,M\,\big(\mathcal{O}((\Delta x)^2) + \mathcal{O}(\Delta T)\big) \\
&\le \bar{T}\,\big(\mathcal{O}((\Delta x)^2) + \mathcal{O}(\Delta T)\big) \\
&= \mathcal{O}((\Delta x)^2) + \mathcal{O}(\Delta T)
\end{aligned}$$

for $j = 0\,,\dots,M-1$. Thus, the order of convergence of the solution z_{j+1} of the finite difference scheme (5.4.28) – (5.4.31) to the solution $\tilde{c}_{j+1}$ to the perturbed parabolic initial–boundary–value problem (5.4.43) is $(2,1)$.

Now, we study the deviation of the solution z_{j+1} of the finite difference scheme (5.4.28) – (5.4.31) to the solution c_{j+1} of the original parabolic initial–boundary–value problem $L\,c = c_T$ with initial condition (5.2.13) and boundary condition (5.4.5). We know that the deviation of the solutions $c(x,T)$ and $\tilde{c}(x,T)$ of the original and the slightly perturbed parabolic initial–boundary–value problem, respectively, is of magnitude $\mathcal{O}((\Delta x)^s)$ for every $(x,T) \in [x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$ by using the weak maximum principle and the accuracy of the approximation $\varphi(x)$ to the original initial condition $\max(1-\exp(x),0)$. Moreover, we just have shown that the convergence of the solution of the finite difference scheme (5.4.28) – (5.4.31) to the solution of the perturbed parabolic initial–boundary–value problem (5.4.43) is of order $(2,1)$.

Using the triangle inequality and these two aspects, we get for the error $c_{j+1} - z_{j+1}$ (note that $N = \frac{x_{\max} - x_{\min}}{\Delta x}$)

$$\begin{aligned} \|c_{j+1} \; - \; z_{j+1}\|_{2,\Delta x} \;&\leq\; \|c_{j+1} \; - \; \tilde{c}_{j+1}\|_{2,\Delta x} \; + \; \|\tilde{c}_{j+1} \; - \; z_{j+1}\|_{2,\Delta x} \\ &= \left(\Delta x \sum_{i=1}^{N-1} (c_{i,j+1} - \tilde{c}_{i,j+1})^2 \right)^{1/2} \; + \; \mathcal{O}((\Delta x)^2) \; + \; \mathcal{O}(\Delta T) \\ &= \left(\Delta x \, (N-1) \, (\mathcal{O}((\Delta x)^s))^2 \right)^{1/2} \; + \; \mathcal{O}((\Delta x)^2) \; + \; \mathcal{O}(\Delta T) \\ &\leq (x_{\max} - x_{\min})^{1/2} \; \mathcal{O}((\Delta x)^s) \; + \; \mathcal{O}((\Delta x)^2) \; + \; \mathcal{O}(\Delta T) \\ &= \mathcal{O}((\Delta x)^{\min\{s\,,2\}}) + \mathcal{O}(\Delta T) \end{aligned}$$

such that the finite difference scheme is convergent of order $(\min\{s\,,2\}\,,1)$ with respect to the discrete L_2–norm $\|\cdot\|_{2,\Delta x}$. ■

In Theorem 5.4.17, we have shown the convergence of the solution of the finite difference scheme (5.4.28) – (5.4.31) to the solution of the parabolic initial–boundary–value problem (5.4.6) by introducing an auxiliary parabolic initial–boundary–value problem and assuming that the accompanying initial condition is given by a sufficiently smooth function $\varphi(x)$ with $\varphi(x_i) = \max(1 - \exp(x_i)\,,0)$, $i = 0\,,\ldots,N$, that approximates the original initial condition $\max(1 - \exp(x)\,,0)$ with a certain accuracy. This result makes clear that the order of this accuracy determines the order of convergence with respect to the spatial variable if $s < 2$. If the solution of the finite difference scheme is directly compared to the solution of the original parabolic initial–boundary–value problem (5.4.6), difficulties appear in the examinations for T tending to 0 because the initial data are not differentiable for $x = 0$. In these cases, finite element methods are often preferred since they possess certain smoothing properties.

Thomée has investigated convergence estimates of spatially and fully discrete schemes resulting from an application of finite element methods on second order parabolic initial–boundary–value problems, see for instance Bramble, Schatz, Thomée, and Wahlbin [12], Huang and Thomée [57], Thomée [104]. We start with presenting convergence estimates for spatially or semidiscrete schemes, i.e. schemes that result from a discretization of the parabolic differential equation with respect to the state space. Let $c_{\Delta x}(T)$ be the solution to this semidiscrete scheme which presents a Δx–dependent finite system of ordinary differential equations in time and let $c(T) = (c(x_1, T), \ldots, c(x_{N-1}, T))$ be the solution of the original parabolic initial–boundary–value problem with x_i, $i = 1, \ldots, N-1$, being fixed by the spatial discretization. For the homogeneous heat equation

$$\begin{aligned} c_T \;&=\; \Delta c \quad \text{in } \Omega \qquad \text{for } T > 0\,, \\ c(\cdot, 0) \;&=\; \varphi \quad\;\; \text{in } \Omega\,, \\ c \;&=\; 0 \quad\;\;\; \text{on } \partial\Omega \quad \text{for } T > 0\,, \end{aligned} \tag{5.4.51}$$

where Ω is a bounded domain in $I\!R^n$ with smooth boundary $\partial\Omega$, Thomée has derived in [104, Theorem 3.5] an error estimate for the spatially discretized system elucidating the relation of the error in the finite element solution and the regularity of the exact solution.

Before we present this result, we briefly introduce the relevant spaces and the corresponding norms. According to the statements of Thomée in [104] we denote by $\|\cdot\|$ the norm in $L_2 = L_2(\Omega)$ and, for a positive integer r, by $\|\cdot\|_r$ that in the Sobolev space $H^r = H^r(\Omega) = W_2^r(\Omega)$. Moreover, for $s \geq 0$, let $\dot{H}^s = \dot{H}^s(\Omega)$ be the subspace of L_2 defined by

$$\dot{H}^s \;=\; \left\{ \varphi = \sum_{k=1}^{\infty} (\varphi\,, v_k)\, v_k \in L_2 \;:\; |\varphi|_s \;=\; \left(\sum_{k=1}^{\infty} \lambda_k^s (\varphi\,, v_k)^2 \right)^{1/2} \;<\; \infty \right\}.$$

with $\{\lambda_k\}_{k=1}^{\infty}$ being the nondecreasing sequence of positive eigenvalues of the eigenvalue problem $-\Delta u = \lambda^2 u$ in Ω and $u = 0$ on $\partial\Omega$, that tends to ∞ with k, and with $\{v_k\}_{k=1}^{\infty}$ being the corresponding sequence of eigenfunctions which form an orthonormal basis in $L_2 = L_2(\Omega)$ [104].

Let us now cite the convergence estimate presented by Thomée in [104, Theorem 3.5]:

Theorem 5.4.18 *Let $\{\mathcal{S}_{\Delta x}\}$ be a family of finite dimensional subspaces of L_2 and $\{\mathcal{A}_{\Delta x}\}$ a family of operators $\mathcal{A}_{\Delta x} : L_2 \to \mathcal{S}_{\Delta x}$, approximating the exact solution operator $\mathcal{A}$ of the Dirichlet problem $-\Delta c = f$ in Ω with $c = 0$ on $\partial\Omega$, such that*

(1) $\mathcal{A}_{\Delta x}$ is selfadjoint, positive semidefinite on L_2, and positive definite on $\mathcal{S}_{\Delta x}$;

(2) $\|(\mathcal{A}_{\Delta x} - \mathcal{A})f\| \;\leq\; k\,(\Delta x)^p\,\|f\|_{p-2}$, for $2 \leq p \leq r$, and $f \in H^{p-2}$.

Moreover, assume that the approximation $\varphi_{\Delta x}$ of the initial condition φ satisfies $\varphi_{\Delta x} = P_{\Delta x}\varphi$ with $P_{\Delta x}$ denoting the orthogonal projection of φ onto $\mathcal{S}_{\Delta x}$ with respect to the inner product in L_2. Then if the initial condition φ fulfils $\varphi \in \dot{H}^s$ and $0 \leq s \leq q \leq r$, we have for the error in the semidiscrete problem

$$\|c_{\Delta x}(T) - c(T)\| \;\leq\; k\,(\Delta x)^q\, T^{-(q-s)/2}\,|\varphi|_s, \quad \textit{for } T > 0\,. \tag{5.4.52}$$

The convergence estimate (5.4.52) shows that a convergence rate of order $\mathcal{O}((\Delta x)^r)$ is satisfied uniformly down to $T = 0$ if the initial data φ are sufficiently smooth, i.e. if $\varphi \in \dot{H}^r$ in case of an accuracy of the approximating solution operator $\mathcal{A}_{\Delta x}$ of order r. If the order of regularity of the initial condition φ is lower, the order of convergence of $\mathcal{O}((\Delta x)^r)$ is only achieved for T bounded away from 0 and the bound depends in a singular way on T as T tends to 0. Note that the above estimate (5.4.52) combines this order of singularity with the smoothness of the initial condition $\varphi(x)$, see Thomée [104, Chap. 3].

Remark 5.4.19 If the initial condition φ of the initial–boundary–value problem (5.4.51) with $\Omega = (0\,,1)$ is given by

$$\varphi(x) \;=\; \begin{cases} -x, & \text{if } x \leq \frac{1}{2} \\ 1 - x, & \text{if } x > \frac{1}{2} \end{cases}$$

then $\varphi \in \dot{H}^{3/2-\epsilon}$ with $\epsilon > 0$ arbitrary small. Thus, in this case the convergence estimate (5.4.52) in Theorem 5.4.18 has the form

$$\|c_{\Delta x}(T) - c(T)\| \;\leq\; k\,(\Delta x)^q\,T^{-(q-\frac{3}{2}+\epsilon)/2}\,|\varphi|_{\frac{3}{2}-\epsilon}$$

for $T > 0$ and $0 \leq \frac{3}{2} - \epsilon \leq q \leq r$. Note that the initial condition (5.2.8) given by $C(\sigma(\cdot,\cdot);K,0) = \max(S-K,0)$ of the parabolic initial–boundary–value problem (5.2.7) with initial condition (5.2.8) and boundary conditions (5.4.2), (5.4.4) in the original variables K and T possesses a similar structure. Hence, we would expect that the smoothness of the initial condition of the parabolic initial–boundary–value problem describing the value of a European call option will be of a similar order.

Huang and Thomée presented in [57] also convergence estimates for semidiscrete schemes that approximate initial–boundary–value problems consisting of a more general parabolic differential equation with the elliptic operator having coefficients depending on both x and T, containing lower order terms and being nonselfadjoint and nonpositive, i.e. a parabolic differential equation of the form

$$c_T \;-\; \sum_{j,k=1}^{n} \frac{\partial}{\partial x_j}\left(a_{jk}\,\frac{\partial c}{\partial x_k}\right) \;+\; \sum_{j=1}^{n} a_j\,\frac{\partial c}{\partial x_j} \;+\; a_0\,c \tag{5.4.53}$$

where $a_{jk}\,, a_j$, and a_0 are smooth functions in $\bar{\Omega} \times [0\,,\bar{T}]$, $a_{jk} = a_{kj}$. In case of nonsmooth initial data they obtained the following result, see Huang and Thomée [57, Theorem 3 and 4]:

Theorem 5.4.20 *Let $\{\mathcal{S}_{\Delta x}\}$ be as in Theorem 5.4.18 and let $\mathcal{A}_{\Delta x}$ be a family of operators $\mathcal{A}_{\Delta x} : L_2 \to \mathcal{S}_{\Delta x}$ approximating the exact solution operator $\mathcal{A}$ of the elliptic part of equation* (5.4.53) *with homogeneous Dirichlet boundary conditions, such that*

(1) $(f\,,\mathcal{A}_{\Delta x}f) \geq 0$ for $f \in L_2$, and $(\chi\,,\mathcal{A}_{\Delta x}\chi) > 0$ for $0 \neq \chi \in \mathcal{S}_{\Delta x}$;

(2) $|(\mathcal{A}_{\Delta x}f\,,g) \;-\; (f\,,\mathcal{A}_{\Delta x}g)| \;\leq\; k\,(f\,,\mathcal{A}_{\Delta x}f)^{1/2}\|\mathcal{A}_{\Delta x}g\|$;

(3) there is an integer $r \geq 2$ such that

$$\|((\tfrac{d}{dt})^j\mathcal{A}_{\Delta x} - (\tfrac{d}{dt})^j\mathcal{A})f\| \;\leq\; k\,(\Delta x)^s\|f\|_{s-2}\,, \quad \text{for } 2 \leq s \leq r\,,\; j = 0\,,1\,;$$

(4) the adjoints of $\mathcal{A}$ and $\mathcal{A}_{\Delta x}$ satisfy for r defined in (3)

$$\|((\tfrac{d}{dt})^j\mathcal{A}^*_{\Delta x} - (\tfrac{d}{dt})^j\mathcal{A}^*)f\| \;\leq\; k\,(\Delta x)^s\|f\|_{s-2}\,, \quad \text{for } 2 \leq s \leq r\,,\; j = 0\,,1\,.$$

Furthermore, assume that the initial condition φ fulfils $\varphi \in L_2$ and that its approximation $\varphi_{\Delta x}$ is given by $\varphi_{\Delta x} = P_{\Delta x}\varphi$ with $P_{\Delta x}$ denoting the orthogonal projection of φ onto $\mathcal{S}_{\Delta x}$ with respect to the inner product in L_2. Then, the error in the semidiscrete problem satisfies

$$\|c_{\Delta x}(T) - c(T)\| \;\leq\; k\,(\Delta x)^r\,T^{-r/2}\,\|\varphi\|, \quad \text{for } T > 0\,. \tag{5.4.54}$$

(Note that if condition (3) in Theorem 5.4.20 is fulfilled with $r = 2$ then the following condition (4) is not necessary to obtain the given error estimate in Theorem 5.4.20, see Huang and Thomée [57].)

Although the result in Theorem 5.4.20 lacks to give a more general relation between the order of singularity and the smoothness of the initial condition φ, it again elucidates that including nonsmooth initial data into the examination of error estimates for semidiscrete schemes introduces singularities in the error estimates for T tending to 0 and that the optimal order of convergence of the solution of the semidiscrete scheme is only achieved for $T > 0$, i.e. for T bounded away from 0.

Finally, we cite an error estimate regarding the maximum–norm for semidiscrete schemes presented by Thomée in [104, Chap. 5]. Restricted to initial–boundary–value problems of the form (5.4.51) and to piecewise linear approximation functions in the finite element method, he showed for nonsmooth initial data the following estimate, see Thomée [104, Theorem 5.4 and 5.5]:

Theorem 5.4.21 *Let $\{\mathcal{S}_{\Delta x}\}$ be a family of finite dimensional subspaces of L_2 and let $\mathcal{P}_{\Delta x}$ be a partition of Ω into disjoint triangles with the union of the triangles determine a polygonal domain $\Omega_{\Delta x} \subset \Omega$. Suppose that $\{\mathcal{S}_{\Delta x}\}$ consists of the continuous functions on the closure $\bar{\Omega}$ of Ω which are linear in each triangle of the triangulation $\mathcal{P}_{\Delta x}$ and which vanish outside of $\Omega_{\Delta x}$. Furthermore, assume that the initial condition φ fulfils $\varphi \in L_2$ and for its approximation $\varphi_{\Delta x}$ holds $\varphi_{\Delta x} = P_{\Delta x}\varphi$ with $P_{\Delta x}$ denoting the orthogonal projection of φ onto $\mathcal{S}_{\Delta x}$. Then, we have for the error in the solution of the semidiscrete problem*

$$\|c_{\Delta x}(T) - c(T)\|_{L_\infty} \;\leq\; k\,(\Delta x)^2\,(\ln(\tfrac{1}{\Delta x}))^4\,T^{-1}\,\|\varphi\|_{L_\infty}, \quad \textit{for } T > 0\,. \tag{5.4.55}$$

Moreover, if for the initial condition φ only holds $\varphi \in L_1$, then we get for the error in the solution of the semidiscrete problem

$$\|c_{\Delta x}(T) - c(T)\|_{L_\infty} \;\leq\; k\,(\Delta x)^2\,(\ln(\tfrac{1}{\Delta x}))^4\,T^{-2}\,\|\varphi\|_{L_1}, \quad \textit{for } T > 0\,. \tag{5.4.56}$$

Note, that although the regularity assumptions on the initial data are weakened in the second part of Theorem 5.4.21, still, an error estimate of order 2 is obtained for T bounded away from 0. Moreover, a logarithmic factor additionally appears in the estimate of the error regarding the maximum–norm. This term, however, might be eliminated according to Thomée, see [104, Chap. 5].

After presenting convergence estimates for spatially semidiscrete problems, we now inspect fully discretized systems for second order parabolic initial–boundary–value problems which are obtained by discretizing the semidiscrete problems with respect to time. We continue to present statements of Thomée represented in [104]. To begin with, let the initial–boundary–value problem be of the form (5.4.51). As in Theorem 5.4.17, the accuracy and the stability of the discretized problems are called on to obtain convergence of the solution of the fully discretized scheme to the solution of the initial–boundary–value problem. For the definition of these two concepts and for the examination of discretization schemes with respect to time, Thomée considers the initial–value problem

$$c' \,+\, \bar{\mathcal{L}}\, c \;=\; 0 \quad \text{for } T > 0, \quad \text{with } c(0) = \varphi\,, \tag{5.4.57}$$

with $\bar{\mathcal{L}}$ being a linear, selfadjoint, positive definite, not necessarily bounded operator with compact inverse, and with the definition domain being a subspace of a separable Hilbert space $\mathcal{M}$. The solution operator of problem (5.4.57) can be written as

$$c(T) = \sum_{j=1}^{N} \, \exp(-\lambda_j T)\,(\varphi\,, v_j)\, v_j\,,$$

with $\{\lambda_j\}_{j=1}^N$ being the eigenvalues of $\bar{\mathcal{L}}$ and $\{v_j\}_{j=1}^N$ denoting a corresponding basis of orthogonal eigenfunctions (with $N \leq \infty$). Thus, the solution operator of this initial–value problem can be viewed as the exponential $\exp(-T\,\bar{\mathcal{L}})$ such that a possible way of defining a single step discrete method for this problem class consists of approximating $\exp(-\lambda)$ by a rational function $r(\lambda)$ that is defined on the spectrum $\sigma(k\,\bar{\mathcal{L}})$ of $k\,\bar{\mathcal{L}}$ [104]. Hence, Thomée [104, Chap. 7] defines the accuracy (order q) and the stability condition of a discretized system

$$z_{j+1} \;=\; r(j\,\bar{\mathcal{L}})\, z_j\,, \quad j \geq 0 \quad \text{and with} \quad z_0 = \varphi$$

by

$$r(\lambda) \;=\; \exp(-\lambda) \;+\; \mathcal{O}(\lambda^{q+1})\,, \quad \text{as } \lambda \to 0$$

and

$$\sup_{\lambda \in \sigma(k\bar{\mathcal{L}})} |r(\lambda)| \leq 1\,,$$

respectively, with $\sigma(\bar{\mathcal{L}}) = \{\lambda_j\}_{j=1}^N$ being the spectrum of $\bar{\mathcal{L}}$. Deriving convergence estimates for nonsmooth initial data a further classification of the rational function $r(\lambda)$ approximating $\exp(-\lambda)$ is needed, see Thomée [104, Chap. 7], i.e. $r(\lambda)$ will be of type I, II, III, or IV, respectively, if

I: $|r(\lambda)| < 1$, for $0 < \lambda < \alpha$, with $\alpha > 0$;

II: $|r(\lambda)| < 1$, for $\lambda > 0$;

III: $|r(\lambda)| < 1$, for $\lambda > 0$, and $|r(\infty)| < 1$;

IV: $|r(\lambda)| < 1$, for $\lambda > 0$, and $r(\infty) = 0$.

If the initial–value problem (5.4.57) represents a spatially semidiscrete problem, Thomée [104, Chap. 7] has introduced an additional classification of $r(\lambda)$. The rational function $r(\lambda)$ is said to be of type I' or II', respectively, if

I': $r(\lambda)$ is of type I and $k\,\lambda_{\max} \leq \alpha_0$, for some α_0 with $0 < \alpha_0 < \alpha$;

II': $r(\lambda)$ is of type II and $k\,\lambda_{\max} \leq \alpha_1$, for some α_1 with $0 < \alpha_1 < \infty$;

with $\lambda_{\max}$ denoting the largest eigenvalue of $\bar{\mathcal{L}}$. Note that the explicit Euler, the implicit Euler and the Crank–Nicolson scheme can be represented in form of a rational function $r(\lambda)$ and that they are of type I, IV and II, respectively.

Analogously to the error estimate of the spatially semidiscrete problem in Theorem 5.4.18, there exists an estimate for the error in the solution of a discrete scheme obtained be discretizing problem (5.4.57) with respect to time. The bound of this error again expresses the relation between the regularity of the data, the order of convergence, and the order of singularity, see Thomée [104, Theorem 7.3]:

Theorem 5.4.22 *Let the discretization scheme*

$$z_{j+1} = r(j\,\bar{\mathcal{L}})z_j\,, \quad \textit{for } j \geq 0 \quad \textit{and with } z_0 = \varphi \tag{5.4.58}$$

be accurate of order q and of type I', II', or III. Moreover, let the initial condition φ fulfil $\varphi \in L_2$. Then, we have for the error in the solution z_j of (5.4.58) *with respect to the solution c of* (5.4.57)

$$\|z_j - c(T_j)\| \;\leq\; k\,(\Delta T)^l\,T_j^{-(l-s)}\,|\varphi|_{2s}, \quad \textit{for } \;\varphi \in \dot{H}^{2s}, \quad 0 \leq s \leq l \leq q\,.$$

Theorem 5.4.22 fits for smooth as well as nonsmooth initial data and its error bound again makes clear that optimal order results holding uniformly down to $T = 0$ are only possible when certain smoothness conditions are fulfilled by the initial data.

Finally, we cite estimates for the error in the solution z_j of fully discretized schemes, i.e. schemes where the discretization is carried out with respect to the state as well as the time variable. Thus, these results express the accuracy of the solution with respect to the discretization sizes of both the state and the time variable. We start with stating the error estimate for nonsmooth initial data with respect to the L_2–norm, see Thomée [104, Theorem 7.7].

Theorem 5.4.23 *Let the time discretization scheme be accurate of order q and of type I', II', or III. Moreover, let $\{\mathcal{S}_{\Delta x}\}$ be a family of finite dimensional subspaces of L_2 and $\{\mathcal{A}_{\Delta x}\}$ a family of operators $\mathcal{A}_{\Delta x} : L_2 \rightarrow \mathcal{S}_{\Delta x}$, approximating the exact solution operator $\mathcal{A}$ of the Dirichlet problem $-\Delta c \;=\; f$ in Ω with $c = 0$ on $\partial\Omega$, such that conditions (1) and (2) given in Theorem 5.4.18 are fulfilled. Furthermore, assume that the initial condition φ satisfies $\varphi \in L_2$ and that its discretized approximation is given by $\varphi_{\Delta x} \;=\; P_{\Delta x}\varphi$ with $P_{\Delta x}$ denoting the orthogonal projection of φ onto $\mathcal{S}_{\Delta x}$ with respect to the inner product in L_2. Then, the error in the solution z_j of a fully discrete scheme satisfies*

$$\|z_j - c_j\| \;\leq\; k\,\left((\Delta x)^r\,T_j^{-r/2} \;+\; (\Delta T)^q\,T_j^{-q}\right)\,\|\varphi\|, \quad \textit{for } \;T_j = j\,\Delta T \;>\; 0\,,$$

Theorem 5.4.23 is a direct consequence of Theorems 5.4.18 and 5.4.22 with $s = 0$ and by using the triangle inequality. Thus, if the initial condition φ fulfils $\varphi \in \dot{H}^{2s}$, then, also the following estimate holds combining the regularity of the data, the order of convergence, and the order of singularity:

$$\|z_j - c_j\| \;\leq\; k\, \left((\Delta x)^p \, T_j^{-(p-s)/2} \, |\varphi|_s \;+\; (\Delta T)^l \, T_j^{-(l-s)} \, |\varphi|_{2s} \right) ,$$

with $0 \leq s \leq p \leq r$ and $0 \leq s \leq l \leq q$.

The error estimate in the maximum–norm is exemplified for a fully discrete scheme for the initial–boundary–value problem (5.4.51) in two spatial variables, with $S_{\Delta x}$ consisting of piecewise linear approximation functions in space that are defined on a quasiuniform triangulations of the spatial domain, see Thomée [104, Theorem 8.6].

Theorem 5.4.24 *Let z_j be defined by the fully discrete scheme*

$$z_j \;=\; r(-k\, \Delta_{\Delta x})^n \, \varphi_{\Delta x}$$

with $r(\lambda)$ being a rational function approximating $\exp(-\lambda)$ that is defined on the spectrum $\sigma(k\,\bar{\mathcal{L}})$ of $k\,\bar{\mathcal{L}}$ with $\bar{\mathcal{L}} = -\Delta_{\Delta x}$ being the discrete Laplacian and let c be the solution to the initial–boundary–value problem (5.4.51)*. Moreover, let the discrete approximation of the initial condition φ be defined by $\varphi_{\Delta x} = P_{\Delta x}\varphi$ with $P_{\Delta x}$ denoting the orthogonal projection of φ onto $\mathcal{S}_{\Delta x}$ with respect to the inner product in L_2. Furthermore, assume that the rational function $r(z)$ approximating $\exp(-\lambda)$ is A–stable, i.e. $|r(z)| \leq 1$ for $|\arg z| \leq \frac{\pi}{2}$, that it satisfies $|r(\infty)| < 1$ and that it is accurate of order q. If $\varphi \in L_\infty$, then, for $T_j > 0$ holds*

$$\|z_j - c_j\|_{L_\infty} \;\leq\; k\, \left((\Delta x)^2 \, (\ln(\tfrac{1}{\Delta x}))^4 \, T_j^{-1} + (\Delta T)^q \, (\ln(\tfrac{1}{\Delta x}))^{2q+4} \, T_j^{-q} \right) \, \|\varphi\|_{L_\infty} .$$

In summary, it may be said, that the convergence estimates in Theorems 5.4.18, 5.4.20 – 5.4.24, which all apply to the solution of finite element schemes, elucidate the difficulties when the initial condition in parabolic initial–boundary–value problems is not sufficiently smooth. In all these results, the error bound contains a singularity for T tending to 0 such that the optimal order of convergence is only achieved for T bounded away from 0.

Although the presented error estimates are derived for an initial–boundary–value problem for the homogeneous heat equation, Thomée mentions that the results of Theorem 5.4.22 and 5.4.23 generalize directly to parabolic equation of the form $c_T \;+\; L\,c \;=\; 0$ where the elliptic operator L is selfadjoint, positive definite, and time independent [104, Chap. 7]. However, since in Theorems 5.4.18 – 5.4.24 only special classes of second–order parabolic initial–boundary–value problems are considered it is possible that convergence estimates for finite element solutions applied to more general second–order parabolic initial–boundary–value problems containing for instance a time–dependent elliptic operator, may even be worst.

5.4.4 Determination of the Discrete Instantaneous Volatility Function

After the introduction of a numerical method for the computation of the European call price function $c(x,T)$, we now return to the construction of the volatility function $\hat{\sigma}(x,T)$. Therefore, we again introduce the dependence of the value of the European call option on the volatility function $\hat{\sigma}(x, T)$ in our notation by denoting the European call price function with $c(\hat{\sigma}(\cdot, \cdot); x, T)$ in the following.

In Subsection 5.3, we have derived that the instantaneous volatility function $\hat{\sigma}(x, T)$ can be implicitly determined by approximating the market European option prices, i.e. we have described the construction of the instantaneous volatility function $\hat{\sigma}(x, T)$ by an inverse problem. In the optimization problem (5.3.2), the value of the European call option is given by a sufficiently smooth real–valued function satisfying the transformed version of the extended Black–Scholes model (5.3.3). Using the finite difference approach (5.4.28) – (5.4.31) introduced in the previous section for numerically solving the parabolic initial–value problem (5.3.3) respective the parabolic initial–boundary–value problem (5.4.6), the European call price function $c(\hat{\sigma}(\cdot, \cdot); x, T)$ is approximated by the finite dimensional solution $z = (z_0\,, z_1\,, \ldots, z_M)$ with $z_j = (z_{0,j}\; \ldots, z_{N,j})^T$, $j = 0, \ldots, M$, of the finite difference scheme (5.4.28) – (5.4.31) (note that we have changed slightly the notation by including the boundary values into the definition of the approximation z). Since for the determination of the approximation z only values of the instantaneous volatility function $\hat{\sigma}(x, T)$ on the grid knots are required, we can replace the functional form $\hat{\sigma}(x, T)$ in problem (5.3.2) by the finite dimensional vector $\bar{\sigma}$ just containing the values of $\hat{\sigma}(x, T)$ at the grid knots (x_i, T_j), $i = 1\,, \ldots, N-1\,,\ j = 0\,, \ldots, M-1$, i.e.

$$
\begin{aligned}
\bar{\sigma} \;&=\; (\hat{\sigma}_0, \ldots, \hat{\sigma}_{M-1}) \in I\!R^{M(N-1)}, \qquad \text{with} \\
\hat{\sigma}_j \;&=\; (\hat{\sigma}_{1,j}, \ldots, \hat{\sigma}_{N-1,j}) \in I\!R^{N-1}\,, \quad j = 0, \ldots M-1\,.
\end{aligned}
$$

In the following, we will denote the solution of the finite difference scheme (5.4.28) – (5.4.31) by $z(\bar{\sigma})$ to elucidate the dependence of z on the volatility parameter $\bar{\sigma}$.

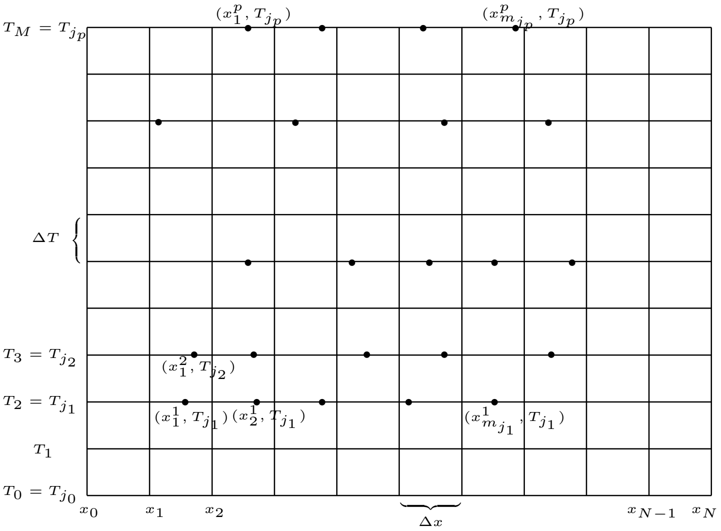


FIGURE 5.3: Arrangement of the market European call options in the finite difference grid

We have already noted above that option prices are quoted in the market for strike prices at discrete intervals and only for a small number of maturities [107]. Furthermore, different series not necessarily contain options with identical strike prices such that, generally, the market

European call options are scattered in the finite difference grid, see Figure 5.3. Moreover, regarding the original variable K the strike prices of market European call options are usually arranged on an equally spaced grid. However, since we have introduced the transformation $x = \ln(\frac{K}{S})$ to eliminate the variable K in the coefficients of the original parabolic differential equation (5.2.7), we cannot guarantee that the transformation of the strike prices x_l, $l = 1, \ldots, m$, of the market European call options, are located on the grid knots in the finite difference mesh. Thus, we cannot guarantee that the corresponding approximations of the market European call options are given by an element $z_{i,j}(\bar{\sigma})$ of the finite difference solution $z(\bar{\sigma})$. However, since the solution $z(\bar{\sigma})$ of the finite difference scheme (5.4.28) – (5.4.31) approximates the solution $c(\hat{\sigma}(\cdot,\cdot); x, T)$ of (5.4.6) with an accuracy of order $(2, 1)$, see Theorem 5.4.17, it is possible to obtain the values of European call options located between two grid points by using polynomial interpolation of second order without disturbing the given order of accuracy of the solution of the finite difference scheme. Thus, let the strike prices $x_l = \ln(\frac{K_l}{S})$ and the maturities T_l of the market European call options satisfy

$$x_{i_l-1} \;\le\; x_l \;\le\; x_{i_l+1} \qquad \text{and} \qquad T_l \;=\; j_l\,\Delta T\,,$$

$l = 1, \ldots, m$, with $i_l \in \{1, \ldots, N-1\}$ and $j_l \in \{0, \ldots, M\}$. Since market European options only exist for a small number of maturities and we have not carried out a transformation with respect to the maturity T, we suppose that the market European call options are located on the chosen time layers in the finite difference grid. Hence, using polynomial interpolation of second order the values of European call options with maturity T_l which are located between the two grid points $(x_{i_l-1}, j_l\,\Delta T)$ and $(x_{i_l+1}, j_l\,\Delta T)$ are given by

$$\begin{aligned} c(x, j_l\Delta T) \;\approx\; & \tfrac{(x-x_{i_l})(x-x_{i_l+1})}{2(\Delta x)^2} z_{i_l-1,j_l}(\bar{\sigma}) + \tfrac{(x-x_{i_l-1})(x_{i_l+1}-x)}{(\Delta x)^2} z_{i_l,j_l}(\bar{\sigma}) + \\ & \tfrac{(x-x_{i_l-1})(x-x_{i_l})}{2(\Delta x)^2} z_{i_l+1,j_l}(\bar{\sigma})\,, \qquad \text{for } x \in [x_{i_l-1}, x_{i_l+1}] \end{aligned} \tag{5.4.59}$$

For a simplified representation of the location of the market European call options in the finite difference grid, we introduce a so–called observation operator that determines the corresponding approximations of the market European call options by being evaluated at the finite dimensional solution $z(\bar{\sigma})$ of the finite difference scheme (5.4.28) – (5.4.31). For the definition of this observation operator, we introduce some notation. Let

$$J \;:=\; \{(K_l, T_l) \;:\; l = 1\,, \ldots, m\}$$

be the set of the pairs of strike prices and maturities of the market European call options and let

$$J_j \;\;:=\;\; \{(K_l, T_l) \in J \;: T_l = j\,\Delta T\}$$

be the set of the pairs of strike prices and maturities of the market European call options where the maturity T_l is assigned to the jth time layer in the finite difference grid with

$$|J_j| = m_j \qquad \text{such that} \qquad \sum_{j=0}^{M} m_j = m\,.$$

Moreover, we assume that there exist p time layers $j_1, j_2, \ldots, j_p \in \{1, \ldots, M\}$ with $m_{j_k} = |J_{j_k}| > 0$ whose indices are united in the index set

$$I = \{j \in \{1, \ldots, M\} : m_j > 0\} \; . \tag{5.4.60}$$

Then,

$$\sum_{j \in I} m_j \; = \; m \qquad \text{and} \qquad |I| \; = \; p \, .$$

We now determine the observation operator $A \in \mathbb{R}^{m \times p(N+1)}$. Since different series of market European call options, i.e. set of market European call options with the same maturity, might not contain options with identical strike prices, we have to determine the location of the market European call options in the finite difference grid separately for the different maturities $j_k \, \Delta T$ with $j_k \in I$, $k = 1\,, \ldots, p$. Let $A_{j_k} = (\alpha_{q,s}^{j_k}) \in \mathbb{R}^{m_{j_k} \times (N+1)}$ be the operator determining the location of the market European call options with maturity $T = j_k \, \Delta T$ in the finite difference grid, $k = 1\,, \ldots\,, p$. Then, we define $A = \operatorname{diag}(A_{j_1}\,, \ldots, A_{j_p})$ as the block diagonal matrix containing the different observation operators A_{j_k} on its diagonal.

Let $K_1^{j_k}\,, \ldots, K_{m_{j_k}}^{j_k}$ be the different strike prices of the market European call options belonging to the series with maturity $T = j_k \, \Delta T$, $k = 1\,, \ldots, p$, with $K_1^{j_k} < \ldots < K_{m_{j_k}}^{j_k}$ without loss of generality. Moreover, define

$$\hat{s}_q^k \; = \; \{\tilde{s} = 0, \ldots, \tfrac{N}{2} - 1 \; : \; x_{2\tilde{s}} \leq \ln(\tfrac{K_q^{j_k}}{S}) \leq x_{2(\tilde{s}+1)}\}$$

for $q = 1, \ldots, m_{j_k}$ and $k = 1\,, \ldots, p$. Considering the polynomial interpolation of second order (5.4.59) for the approximation of European call option values that are located between two grid points, then, the elements $\alpha_{q,s}^{j_k}$, $q = 1, \ldots, m_{j_k}\,,\; s = 0, \ldots, N$ of the observation operator A_{j_k}, $k = 1\,, \ldots, p$, are given by

$$\alpha_{q,s}^{j_k} = \begin{cases} \frac{(x_q^k - x_{s+1})(x_q^k - x_{s+2})}{2(\Delta x)^2} & \text{if } s = 2\,\hat{s}_q^k \\[2ex] \frac{(x_q^k - x_{s-1})(x_{s+1} - x_q^k)}{(\Delta x)^2} & \text{if } s = 2\,\hat{s}_q^k + 1 \\[2ex] \frac{(x_q^k - x_{s-2})(x_q^k - x_{s-1})}{2(\Delta x)^2} & \text{if } s = 2\,\hat{s}_q^k + 2 \\[2ex] 0 & \text{otherwise} \end{cases}$$

with $x_i^k = \ln(\frac{K_i^{j_k}}{S})$, $i = 1, \ldots, m_{j_k}$, and $x_s = x_{\min} + s\,\Delta x$.

Let $\hat{z}(\bar{\sigma}) = (z_{j_1}(\bar{\sigma})\,, \ldots, z_{j_p}(\bar{\sigma}))^T \in \mathbb{R}^{p(N+1)}$ with $j_k \in I$, $k = 1\,, \ldots, p$, then, an evaluation of the observation operator A at $\hat{z}(\bar{\sigma})$, i.e.

$$z_{bs}(\bar{\sigma}) \; = \; A\,\hat{z}(\bar{\sigma}) \; = \; \big(A_{j_1}\, z_{j_1}(\bar{\sigma})\,, A_{j_2}\, z_{j_2}(\bar{\sigma})\,, \ldots, A_{j_p}\, z_{j_p}(\bar{\sigma})\big)^T$$

causes that $z_{bs}(\bar{\sigma})$ contains the computed European call option values using the finite difference scheme (5.4.28) – (5.4.31) for the solution of the transformed version of the extended Black–Scholes equation (5.3.3) that correspond to the market European call options. Furthermore, if we set

$$\hat{C}_{j_k} = \left(\hat{C}_{K_1^{j_k}, T_1^{j_k}}, \ldots, \hat{C}_{K^{j_k}_{m_{j_k}}, T^{j_k}_{m_{j_k}}} \right)^T \in I\!R^{m_{j_k}} ,$$

i.e. $\hat{C}_{j_k}$ contains the values of the market European call options with maturity $T = j_k \Delta T$, $k = 1, \ldots, p$, and

$$c_{observ} = \left(\hat{C}_{j_1}, \hat{C}_{j_2}, \ldots, \hat{C}_{j_p} \right)$$

and define

$$R(\bar{\sigma}) = z_{bs}(\bar{\sigma}) - c_{observ} \tag{5.4.61}$$

then

$$\min_{\bar{\sigma} \in I\!R^n} \phi(\bar{\sigma}) \;=\; \min_{\bar{\sigma} \in I\!R^n} \tfrac{1}{2} \| R(\bar{\sigma}) \|^2 \tag{5.4.62}$$

with

$$n \;=\; M(N-1)$$

($\| \cdot \|$ is the Euclidean norm in $I\!R^{M(N-1)}$) is the discrete counterpart of optimization problem (5.3.2) describing the construction of the instantaneous volatility function $\hat{\sigma}(x,T)$ if we ignore the smoothness condition on $\hat{\sigma}(x,T)$ for a moment.

Since the residual function R depends nonlinearly on the finite dimensional version of the instantaneous volatility function $\bar{\sigma}$, problem (5.4.62) represents a nonlinear least–squares problem. Moreover, note that usually the number of market European call options is small and a sufficient accuracy is only achieved with a certain number of discretization knots such that, in general, these problems will be underdetermined, i.e. $n \gg m$. Since the degree of freedom of these problems is too large, they typically have an infinite number of solutions so that the ill–posedness of the optimization problem (5.3.2) is carried over to its finite dimensional counterpart (5.4.62).

As already mentioned in earlier chapters, in the case of underdetermined nonlinear least–squares problem it is likely that the residual function R at the solution is small. Hence, for the solution of problem (5.4.62) we use the Inexact Gauss–Newton Algorithm 2.2.3 presented in Chapter 2. This algorithm has the advantage that on the one side in case of a zero residual at the solution the algorithm exhibits good asymptotic convergence behavior and on the other side by using a trust–region strategy for the solution of the Gauss–Newton subproblems a regularization is imposed on the subproblems that overcomes their ill-posedness, see Theorem 2.2.2.

If the Euclidean norm is used in the definition of the trust–region $\mathcal{U}_i$ given by (2.2.6), then the restriction in the Gauss–Newton subproblems prevents the updates of the iterates from becoming too large. In the context of constructing the instantaneous volatility function $\hat{\sigma}(x,T)$, however, one wishes to determine the volatility function so that the extended Black–Scholes model calibrates the values of the market European call options and which shows the most smooth behavior, i.e. $\| \nabla \hat{\sigma}(x,T) \|$ should be as small as possible. Thus, in this case, regularization is carried out with respect to the smoothness of the volatility function. For the definition of the trust–region, this means that the norm of the discrete version of the differential operator evaluated at the updates of the iterates should be bounded.

Hence for the definition of this trust–region, we first of all need to make the discrete version of $\nabla\hat{\sigma}(x,T)$ available.

Lemma 5.4.25 *Let $\hat{\sigma}(x,T) : I\!R \times [0,\bar{T}] \to I\!R$ be the instantaneous volatility function. Using the finite grid generated by the finite difference scheme* (5.4.28) – (5.4.31) *and approximating the partial derivatives of $\hat{\sigma}(x,T)$ with a forward difference quotient, then a discrete version of the Jacobian $\nabla\hat{\sigma}(x,T)$ is given by*

$$D\,\bar{\sigma} \;=\; \begin{pmatrix} D_x \\ D_T \end{pmatrix} \cdot \bar{\sigma} \tag{5.4.63}$$

with

$$D_x = \tfrac{1}{\Delta x}\begin{pmatrix} E & & & \\ & E & & \\ & & \ddots & \\ & & & E \end{pmatrix} \quad \text{and} \quad D_T = \tfrac{1}{\Delta T}\begin{pmatrix} -I & I & & 0 \\ & -I & I & \\ & & \ddots & \ddots \\ 0 & & -I & I \end{pmatrix},$$

$D_x \in I\!R^{(N-2)M\times(N-1)M}$, $D_T \in I\!R^{(N-1)(M-1)\times(N-1)M}$ and where $I \in I\!R^{(N-1)\times(N-1)}$ is the $(N-1)\times(N-1)$ identity matrix and

$$E = \begin{pmatrix} -1 & 1 & & 0 \\ & -1 & 1 & \\ & & \ddots & \ddots \\ 0 & & -1 & 1 \end{pmatrix} \in I\!R^{(N-2)\times(N-1)}.$$

Proof: Since the instantaneous volatility function $\hat{\sigma}(x,T)$ is of the form $\hat{\sigma} : I\!R \times [0,\bar{T}] \to I\!R$ the Jacobian of $\hat{\sigma}(x,T)$ has the form

$$\nabla\hat{\sigma}(x,T) = \left(\tfrac{\partial}{\partial x}\hat{\sigma}(x,T)\,,\,\tfrac{\partial}{\partial T}\hat{\sigma}(x,T)\right)\,.$$

For the derivation of the discrete form of $\nabla\hat{\sigma}(x,T)$ we use the same discretization of the state and the time space than in the finite difference scheme (5.4.28) – (5.4.31), i.e.

$$\begin{aligned} x_i &= x_{\min} + i\,\Delta x\,, \quad i = 0,\ldots,N\,, \\ T_j &= j\,\Delta T\,, \qquad\qquad j = 0,\ldots,M\,, \end{aligned}$$

with $\Delta x = (x_{\max} - x_{\min})/N$ and $\Delta T = \bar{T}/M$. Furthermore, we set

$$\begin{aligned} \nabla_x\hat{\sigma}_{i,j} &= \tfrac{\partial}{\partial x}\hat{\sigma}(x_i,T_j)\,, \quad i = 1,\ldots,N-2\,,\; j = 0,\ldots,M-1\,, \\ \nabla_T\hat{\sigma}_{i,j} &= \tfrac{\partial}{\partial T}\hat{\sigma}(x_i,T_j)\,, \quad i = 1,\ldots,N-1\,,\; j = 0,\ldots,M-2\,. \end{aligned}$$

First, we examine the partial derivative of $\hat{\sigma}(x,T)$ with respect to x. A discretized version of $\frac{\partial}{\partial x}\hat{\sigma}(x,T)$ is given by

$$\nabla_x\bar{\sigma} = (\nabla_x\hat{\sigma}_0, \nabla_x\hat{\sigma}_1, \dots, \nabla_x\hat{\sigma}_{M-1})^T \in I\!R^{(N-2)M}$$

with

$$\nabla_x\hat{\sigma}_j = (\nabla_x\hat{\sigma}_{1,j}, \nabla_x\hat{\sigma}_{2,j}, \dots, \nabla_x\hat{\sigma}_{N-2,j})^T \in I\!R^{N-2}\,, \quad j = 0,\dots,M-1\,,$$

i.e. $\nabla_x\bar{\sigma}$ contains the partial derivative $\frac{\partial}{\partial x}\hat{\sigma}(x,T)$ evaluated at the different grid knots in the finite difference grid. We approximate the partial derivative of $\hat{\sigma}(x,T)$ with respect to x by using the forward difference quotient, i.e.

$$\tfrac{\partial}{\partial x}\hat{\sigma}(x_i,T_j) = \nabla_x\hat{\sigma}_{i,j} \approx \frac{\hat{\sigma}_{i+1,j} - \hat{\sigma}_{i,j}}{\Delta x}\,, \quad i = 1,\dots,N-2\,,\ j = 0,\dots,M-1\,.$$

Let

$$E = \begin{pmatrix} -1 & 1 & & 0 \\ & -1 & 1 & \\ & & \ddots & \ddots \\ & 0 & & -1 & 1 \end{pmatrix} \in I\!R^{(N-2)\times(N-1)},$$

then, an approximation of the partial derivative $\frac{\partial}{\partial x}\hat{\sigma}(x,T)$ evaluated at the different grid knots in the jth time layer is given by

$$\tfrac{1}{\Delta x}\, E\,\hat{\sigma}_j \;\approx\; \nabla_x\hat{\sigma}_j\,, \quad j = 0,\dots,M-1,$$

and

$$\underbrace{\frac{1}{\Delta x}\begin{pmatrix} E & & & \\ & E & & \\ & & \ddots & \\ & & & E \end{pmatrix}}_{=:\,D_x} \cdot \begin{pmatrix} \hat{\sigma}_0 \\ \vdots \\ \hat{\sigma}_{M-1} \end{pmatrix} \approx \begin{pmatrix} \nabla_x\,\hat{\sigma}_0 \\ \vdots \\ \nabla_x\,\hat{\sigma}_{M-1} \end{pmatrix}$$

with $D_x \in I\!R^{(N-2)M\times(N-1)M}$. The rank of the discrete differentiation operator D_x is given by $\operatorname{rank} D_x = (N-2)M$ since $\operatorname{rank} E = N-2$ and $D_x = \operatorname{diag}(E,\dots,E)$ with E appearing M times on the diagonal of D_x according to the M time steps.

Similarly to $\frac{\partial}{\partial x}\hat{\sigma}$, a discrete version of the partial derivative of $\hat{\sigma}(x,T)$ with respect to the maturity T is given by

$$\nabla_T\,\bar{\sigma} = (\nabla_T\,\hat{\sigma}_0, \nabla_T\,\hat{\sigma}_1, \dots, \nabla_T\,\hat{\sigma}_{M-2})^T \in I\!R^{(N-1)(M-1)}$$

with

$$\nabla_T\,\hat{\sigma}_j = (\nabla_T\,\hat{\sigma}_{1,j}, \nabla_T\,\hat{\sigma}_{2,j}, \dots, \nabla_T\,\hat{\sigma}_{N-1,j})^T \in I\!R^{N-1}\,, \quad j = 0,\dots,M-2\,.$$

Using the forward difference quotient for the approximation of the partial derivative of $\hat{\sigma}(x_i,T_j)$ with respect to T yields

$$\tfrac{\partial}{\partial T}\hat{\sigma}(x_i,T_j) = \nabla_T\,\hat{\sigma}_{i,j} \approx \frac{\hat{\sigma}_{i,j+1} - \hat{\sigma}_{i,j}}{\Delta T}\,, \quad i = 1,\dots,N-1\,,\ j = 0,\dots,M-2\,.$$

Hence

$$\underbrace{\frac{1}{\Delta T}\begin{pmatrix} -I & I & & 0 \\ & -I & I & \\ & & \ddots & \ddots \\ & 0 & & -I & I \end{pmatrix}}_{=:\,D_T} \cdot \begin{pmatrix} \hat{\sigma}_0 \\ \hat{\sigma}_1 \\ \vdots \\ \hat{\sigma}_{M-1} \end{pmatrix} \approx \begin{pmatrix} \nabla_T\,\hat{\sigma}_0 \\ \nabla_T\,\hat{\sigma}_1 \\ \vdots \\ \nabla_T\,\hat{\sigma}_{M-2} \end{pmatrix}$$

with $I \in I\!R^{(N-1)\times(N-1)}$ is the $(N-1)\times(N-1)$ identity matrix, $D_T \in I\!R^{(N-1)(M-1)\times(N-1)M}$ and rank $D_T = (N-1)(M-1)$ since D_T is an upper triangular matrix with non–zero diagonal elements.

Thus, a discrete version of the Jacobian $\nabla\hat{\sigma}(x,T)$ of the volatility function $\hat{\sigma}(x,T)$ is given by

$$\nabla\bar{\sigma} \;=\; (\nabla_x\,\hat{\sigma}_0,\ldots,\nabla_x\,\hat{\sigma}_{M-1}\,,\nabla_T\,\hat{\sigma}_0,\ldots,\nabla_T\,\hat{\sigma}_{M-2})^T \;\approx\; \underbrace{\begin{pmatrix} D_x \\ D_T \end{pmatrix}}_{=:\,D} \cdot\bar{\sigma}$$

with $D \in I\!R^{\tilde{m}\times(N-1)M}$ and $\tilde{m} = (N-2)M + (N-1)(M-1)$ such that Lemma 5.4.25 is proven. ■

Remark 5.4.26 Since $D_x\,\bar{\sigma} = 0$ if and only if $\bar{\sigma}$ satisfies

$$\hat{\sigma}_{i+1,j} \;=\; \hat{\sigma}_{i,j}\,, \qquad i = 1,\ldots,N-2\,,\; j = 1,\ldots,M-1\,, \tag{5.4.64}$$

and $D_T\,\bar{\sigma} = 0$ if and only if $\bar{\sigma}$ fulfils

$$\hat{\sigma}_{i,j+1} \;=\; \hat{\sigma}_{i,j}\,, \qquad i = 1,\ldots,N-1\,,\; j = 1,\ldots,M-2\,, \tag{5.4.65}$$

the homogeneous linear system $D\bar{\sigma} = 0$ is fulfilled if and only if $\bar{\sigma}$ satisfies simultaneously (5.4.64) and (5.4.65). However, this is only the case if $\bar{\sigma} = \lambda\,e$, with $e = (1,\ldots,1)^T$, $e \in I\!R^{(N-1)M}$ and $\lambda \in I\!R$ such that the null–space $\mathcal{N}(D)$ of the discrete differentiation operator D is given by

$$\mathcal{N}(D) = \left\{\bar{\sigma} \in I\!R^{(N-1)M} \;:\; \bar{\sigma} = \lambda e\,, \lambda \in I\!R\right\}\,.$$

Thus, the null–space of the discrete version of the differentiation operator ∇ corresponds to the null–space of its continuous counterpart.

Now, to achieve regularization regarding smoothness of the instantaneous volatility function $\hat{\sigma}(x,T)$, the trust–region in the Inexact Gauss–Newton Algorithm 2.2.3 might be defined by setting

$$\|D\,\delta\bar{\sigma}\|_2 \leq \Delta\,, \tag{5.4.66}$$

i.e. that the norm of the update $\delta\bar{\sigma}$ of the current iterate $\bar{\sigma}$ scaled with the discretized differentiation operator has to be bounded by Δ. Since according to Remark 5.4.26 the matrix D has a nontrivial null space

$$\|x\|_{D^TD} = \|D\,x\|_2 = \sqrt{x^T D^T D\,x}$$

is only a seminorm since for all $\bar{x} \in \mathcal{N}(D)$ we have that $\|\bar{x}\|_{D^TD} = 0$. Thus, using condition (5.4.66) for the definition of the trust–region is not sufficient for obtaining well–posed Gauss–Newton subproblems. Since $J(\bar{\sigma})^T J(\bar{\sigma}) + \lambda\, D^T D$ ($J(\bar{\sigma})$ is the Jacobian of the residual functions $R(\bar{\sigma})$) might have a nontrivial null–space, it is possible that the solution of the Gauss–Newton subproblem is not unique, see Theorem 2.2.2.

Moreover, if the trust–region $\mathcal{U}_i$ (see (2.2.6)) is defined with respect to the seminorm $\|\cdot\|_{D^TD}$ it is possible that the solution to the Gauss–Newton subproblem is not located in a neighborhood of the current iterate $\bar{\sigma}$. The linearised problem in the Gauss–Newton method is derived from the nonlinear problem by using Taylor approximation which only can be trusted in a neighborhood of the expansion point. To ensure that the solution of the Gauss–Newton subproblem is restricted to a certain neighborhood of the current Gauss–Newton iterate $\bar{\sigma}$, additionally, we have to guarantee that the norm of the solution of these subproblems is not getting too large, i.e. $\|\delta\bar{\sigma}\|_2 < \Delta$. Hence, we additionally introduce the restriction that the norm of the orthogonal projection of the iterate $\delta\bar{\sigma}$ onto the null–space $\mathcal{N}(D)$ of D is bounded.

Including this additional restriction into the definition of the trust–region, the Gauss–Newton subproblems become

$$\begin{array}{c} \min\limits_{\delta\bar{\sigma}\in I\!R^n} \langle J(\bar{\sigma})^T R(\bar{\sigma}), \delta\bar{\sigma}\rangle + \frac{1}{2}\langle \delta\bar{\sigma}\,, J(\bar{\sigma})^T J(\bar{\sigma})\,\delta\bar{\sigma}\rangle \\ \text{s.t.} \quad \|D\delta\bar{\sigma}\|_2 \le \bar{\Delta} \quad \text{and} \quad \|P_{\mathcal{N}(D)}\delta\bar{\sigma}\|_2 \le \tilde{\Delta} \end{array} \tag{5.4.67}$$

with the orthogonal projection $P_{\mathcal{N}(D)}$ onto $\mathcal{N}(D)$ being given by

$$P_{\mathcal{N}(D)} = \frac{1}{\|e\|_2^2}\, e\, e^T = \frac{1}{n}\, e\, e^T$$

where $e = (1,\ldots,1)^T \in I\!R^n$.

In the following, we combine the two restrictions in (5.4.67) by defining

$$\|\delta\bar{\sigma}\|_{\bar{D}} = \sqrt{\langle \delta\bar{\sigma}\,, \bar{D}\,\delta\bar{\sigma}\rangle} \tag{5.4.68}$$

with $\bar{D}$ is given by

$$\bar{D} = D^T D + P_{\mathcal{N}(D)}^T P_{\mathcal{N}(D)}\,.$$

Lemma 5.4.27 *Let D be the discrete version of the differentiation operator ∇ given in Lemma 5.4.25 and let $P_{\mathcal{N}(D)}$ be the orthogonal projection onto the null–space $\mathcal{N}(D)$ of D. Then, $\|\cdot\|_{\bar{D}}$, defined in* (5.4.68)*, represents a norm.*

Proof: Since

$$\begin{aligned} \langle \delta\bar{\sigma}\,, \bar{D}\,\delta\bar{\sigma}\rangle &= \langle \delta\bar{\sigma}\,, (D^T D + P_{\mathcal{N}(D)}^T P_{\mathcal{N}(D)})\,\delta\bar{\sigma}\rangle \\ &= \langle \delta\bar{\sigma}\,, D^T D\,\delta\bar{\sigma}\rangle + \langle \delta\bar{\sigma}\,, P_{\mathcal{N}(D)}^T P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\rangle \\ &= \|D\,\delta\bar{\sigma}\|_2^2 + \|P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\|_2^2 \\ &\ge 0 \end{aligned}$$

the square root in (5.4.68) is defined and $\|\delta\bar{\sigma}\|_{\bar{D}} \geq 0$. Furthermore, $\|\delta\bar{\sigma}\|_{\bar{D}} = 0$ if and only if $\delta\bar{\sigma} = 0$ since $\|\delta\bar{\sigma}\|_{\bar{D}} = 0$ if and only if $\|D\,\delta\bar{\sigma}\|_2^2 = 0$ and $\|P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\|_2^2 = 0$. The former term is zero if and only if $\delta\bar{\sigma} \in \mathcal{N}(D)$. In case $\delta\bar{\sigma} \in \mathcal{N}(D)$ and $\delta\bar{\sigma} \neq 0$ it is

$$P_{\mathcal{N}(D)}\,\delta\bar{\sigma} = \tfrac{e^T\delta\bar{\sigma}}{n}\,e \neq 0$$

since $e^T\delta\bar{\sigma} \neq 0$ because $\delta\bar{\sigma} \in \mathcal{N}(D)$. Hence,

$$\|D\,\delta\bar{\sigma}\|_2^2 + \|P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\|_2^2 \neq 0\,.$$

If, on the other side, $\|P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\|_2^2 = 0$ and $\delta\bar{\sigma} \neq 0$ this means that $e^T\delta\bar{\sigma} = 0$, i.e. $\delta\bar{\sigma}$ is orthogonal to $\mathcal{N}(D)$. However, then $D\,\delta\bar{\sigma} \neq 0$. Hence, $\|D\,\delta\bar{\sigma}\|_2^2$ and $\|P_{\mathcal{N}(D)}\,\delta\bar{\sigma}\|_2^2$ are simultaneously equal to 0 if and only if $\delta\bar{\sigma} = 0$ such that $\|\delta\bar{\sigma}\|_{\bar{D}} = 0$ if and only if $\delta\bar{\sigma} = 0$.

Moreover,

$$\begin{aligned}
\|\delta\bar{\sigma}_1 + \delta\bar{\sigma}_2\|_{\bar{D}}^2 &= \langle\delta\bar{\sigma}_1 + \delta\bar{\sigma}_2\,, \bar{D}\,(\delta\bar{\sigma}_1 + \delta\bar{\sigma}_2)\rangle \\
&= \langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle + 2\,\langle\delta\bar{\sigma}_1, \bar{D}\,\delta\bar{\sigma}_2\rangle + \langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle\,.
\end{aligned} \tag{5.4.69}$$

Since $\bar{D}$ is symmetric and positive definite it exists the square root of $\bar{D}$ such that $\bar{D} = \bar{D}^{1/2}\bar{D}^{1/2}$ with $\bar{D}^{1/2}$ being symmetric and positive definite. Hence, by using the Cauchy–Schwarz inequality, we get

$$\begin{aligned}
\langle\delta\bar{\sigma}_1, \bar{D}\,\delta\bar{\sigma}_2\rangle &= \langle\delta\bar{\sigma}_1, \bar{D}^{1/2}\bar{D}^{1/2}\,\delta\bar{\sigma}_2\rangle \\
&= \langle\bar{D}^{1/2}\,\delta\bar{\sigma}_1, \bar{D}^{1/2}\,\delta\bar{\sigma}_2\rangle \\
&\leq \|\bar{D}^{1/2}\,\delta\bar{\sigma}_1\|_2\,\|\bar{D}^{1/2}\,\delta\bar{\sigma}_2\|_2 \\
&= \sqrt{\langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle}\sqrt{\langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle}\,.
\end{aligned}$$

Including this result into (5.4.69) we get

$$\begin{aligned}
\|\delta\bar{\sigma}_1 + \delta\bar{\sigma}_2\|_{\bar{D}}^2 &= \langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle + 2\,\langle\delta\bar{\sigma}_1, \bar{D}\,\delta\bar{\sigma}_2\rangle + \langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle \\
&\leq \langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle + 2\,\sqrt{\langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle}\sqrt{\langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle} + \langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle \\
&= \left(\sqrt{\langle\delta\bar{\sigma}_1\,, \bar{D}\,\delta\bar{\sigma}_1\rangle} + \sqrt{\langle\delta\bar{\sigma}_2\,, \bar{D}\,\delta\bar{\sigma}_2\rangle}\right)^2 \\
&= (\|\delta\bar{\sigma}_1\|_{\bar{D}} + \|\delta\bar{\sigma}_2\|_{\bar{D}})^2
\end{aligned}$$

which verifies the triangle inequality for $\|\cdot\|_{\bar{D}}$. Finally,

$$\begin{aligned}
\|\lambda\,\delta\bar{\sigma}\|_{\bar{D}}^2 &= \langle\lambda\,\delta\bar{\sigma}\,, \bar{D}\,\lambda\,\delta\bar{\sigma}\rangle \\
&= \lambda^2\,\langle\delta\bar{\sigma}\,, \bar{D}\,\delta\bar{\sigma}\rangle \\
&= \lambda^2\,\|\delta\bar{\sigma}\|_{\bar{D}}^2
\end{aligned}$$

such that $\|\cdot\|_{\bar{D}}$ satisfies all attributes of a norm. ■

Now, using the norm $\|\cdot\|_{\bar{D}}$ for the restriction of the iterates $\delta\bar{\sigma}$ the subproblems in the Inexact Gauss–Newton Algorithm 2.2.3 have the form

$$\begin{aligned}&\min_{\delta\bar{\sigma}\in I\!R^n} \langle J(\bar{\sigma})^T R(\bar{\sigma}), \delta\bar{\sigma}\rangle + \tfrac{1}{2}\langle \delta\bar{\sigma}\,, J(\bar{\sigma})^T J(\bar{\sigma})\,\delta\bar{\sigma}\rangle \\ &\text{s.t.} \quad \|\delta\bar{\sigma}\|_{\bar{D}} \leq \Delta\end{aligned} \tag{5.4.70}$$

with $J(\bar{\sigma})$ being the Jacobian of the residual function $R(\bar{\sigma})$. These subproblems are now well–posed according to Theorem 2.2.2.

Usually, a weighted norm in the definition of the trust–region is introduced to obtain a better scaling of the variables in the Gauss–Newton subproblem. Note, however, that the introduction of the norm $\|\cdot\|_{\bar{D}}$ in the subproblems (5.4.70) is originated by the characteristic features of the given problem describing the construction of the instantaneous volatility parameter $\bar{\sigma}$.

5.5 Evaluation of the Jacobian

Using the Inexact Gauss–Newton Algorithm 2.2.3 for the solution of the underdetermined nonlinear least–squares problem (5.4.62) in each iteration of the Gauss–Newton method, we solve the resulting restricted linear least–squares problem

$$\begin{aligned}&\min_{\delta\bar{\sigma}\in I\!R^n} \langle J(\bar{\sigma})^T R(\bar{\sigma}), \delta\bar{\sigma}\rangle + \tfrac{1}{2}\langle \delta\bar{\sigma}\,, J(\bar{\sigma})^T J(\bar{\sigma})\,\delta\bar{\sigma}\rangle \\ &\text{s.t.} \quad \|\delta\bar{\sigma}\|_{\bar{D}} \leq \Delta\end{aligned}$$

appearing in every iteration of the Gauss–Newton method with the Modified Preconditioned Conjugate Gradient Method 2.2.2. We have seen in Chapter 2 that the conjugate gradient method requires an efficient evaluation method of $J(\bar{\sigma})\delta\bar{\sigma}$ as well as $J(\bar{\sigma})^T\delta y$. In the two following theorems, we show how this can be achieved without saving the matrix $J(\bar{\sigma})$. For the proof of the first of these two theorems, the following lemma is useful.

Lemma 5.5.1 *Let* $f : I\!R^q \to I\!R^n$ *and* $W : I\!R^q \to I\!R^{m\times n}$ *with*

$$\begin{aligned} f(y) &= (f_1(y), f_2(y), \ldots, f_n(y))^T \in I\!R^n\,, \\ W(y) &= \{w_{i,j}(y)\} = (w_1(y), w_2(y), \ldots, w_n(y)) \in I\!R^{m\times n}\end{aligned}$$

where $f_j(y) \in I\!R$, $w_j(y) = (w_{1,j}(y), w_{2,j}(y), \ldots, w_{m,j}(y))^T \in I\!R^m$, $y = (y_1, \ldots, y_q) \in I\!R^q$ *and* $f\,, w_j \in C^1(I\!R^q)$, $j = 1, \ldots, n$. *Then, for* $\delta y \in I\!R^q$,

$$\tfrac{d}{dy}\left\{W(y)f(y)\right\}\delta y = \sum_{l=1}^{n} f_l(y)\tfrac{d}{dy}w_l(y)\delta y + W(y)\tfrac{d}{dy}f(y)\delta y \in I\!R^m$$

Proof: Since

$$W(y)f(y) \;=\; f_1(y)w_1(y) + \ldots + f_n(y)w_n(y) \;=\; \sum_{l=1}^{n} f_l(y)w_l(y)$$

Lemma 5.5.1 follows immediately by applying the product rule to $\frac{d}{dy}\,(f_l(y)w_l(y))$. ■

In Subsection 5.4.2, we have shown that approximations of the value of a European call option can be computed recursively by using the finite difference scheme (5.4.28), i.e.

$$(I + \theta\tilde{D}_j(\bar{\sigma}))\, z_{j+1}(\bar{\sigma}) \;=\; (I - (1-\theta)\tilde{D}_j(\bar{\sigma}))\, z_j(\bar{\sigma}) \;+\; \tilde{B}_j(\bar{\sigma})\,,$$

where we include the dependency of the matrices $\tilde{D}_j$ and the vectors $\tilde{B}_j$, $j = 0, \ldots, M-1$, on $\bar{\sigma}$ by writing

$$\begin{aligned} \tilde{D}_j &= \tilde{D}_j(\bar{\sigma})\,, \\ \tilde{B}_j &= \tilde{B}_j(\bar{\sigma})\,. \end{aligned}$$

For simplicity, we define

$$\begin{aligned} U_j(\bar{\sigma}) &= I + \theta\tilde{D}_j(\bar{\sigma}) && \in \mathbb{R}^{(N-1)\times(N-1)}\,, && j = 0, \ldots, M-1\,, \\ V_j(\bar{\sigma}) &= I - (1-\theta)\tilde{D}_j(\bar{\sigma}) && \in \mathbb{R}^{(N-1)\times(N-1)}\,, && j = 0, \ldots, M-1\,, \end{aligned}$$

with $\tilde{D}_j(\bar{\sigma})$ defined in (5.4.29). Furthermore, let the conditions of Theorem 5.4.3 or Corollary 5.4.5 are satisfied so that $U_j(\bar{\sigma})$ is nonsingular for $j = 0, \ldots, M-1$. Let $U_l^j(\bar{\sigma})$ and $V_l^j(\bar{\sigma})$ be the lth column of $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$, respectively, $l = 1, \ldots, N-1$, $j = 0, \ldots, M-1$, then

$$\begin{aligned} U_l^j(\bar{\sigma}) &= e_l + \theta\tilde{D}_l^j(\bar{\sigma}) && \in \mathbb{R}^{(N-1)} \\ V_l^j(\bar{\sigma}) &= e_l - (1-\theta)\tilde{D}_l^j(\bar{\sigma}) && \in \mathbb{R}^{(N-1)} \end{aligned} \qquad (5.5.1)$$

where e_l is the lth Euclidean unit vector in $\mathbb{R}^{(N-1)}$ and $\tilde{D}_l^j(\bar{\sigma})$ is the lth column of the matrix $\tilde{D}_j(\bar{\sigma})$, i.e. for $l = 2, \ldots, N-2$

$$\tilde{D}_l^j(\bar{\sigma}) = \alpha\,(0, \ldots, 0, u_{l-1}^j(\bar{\sigma}), m_l^j(\bar{\sigma}), v_{l+1}^j(\bar{\sigma}), 0, \ldots, 0)^T \in \mathbb{R}^{(N-1)},$$

and

$$\begin{aligned} \tilde{D}_1^j(\bar{\sigma}) &= \alpha\,(m_1^j(\bar{\sigma}), v_2^j(\bar{\sigma}), 0, \ldots, 0)^T && \in \mathbb{R}^{(N-1)}\,, \\ \tilde{D}_{N-1}^j(\bar{\sigma}) &= \alpha\,(0, \ldots, 0, u_{N-2}^j(\bar{\sigma}), m_{N-1}^j(\bar{\sigma}))^T && \in \mathbb{R}^{(N-1)} \end{aligned}$$

with $v_i^j(\bar{\sigma})$, $m_i^j(\bar{\sigma})$ and $u_i^j(\bar{\sigma})$ defined in (5.4.19).

We now show that the Jacobian of

$$R(\bar{\sigma}) = \begin{pmatrix} A_{j_1}\, z_{j_1}(\bar{\sigma}) - \hat{C}_{j_1} \\ A_{j_2}\, z_{j_2}(\bar{\sigma}) - \hat{C}_{j_2} \\ \vdots \\ A_{j_p}\, z_{j_p}(\bar{\sigma}) - \hat{C}_{j_p} \end{pmatrix}$$

(for the definition of A_{j_i} and $\hat{C}_{j_i}$, $i = 1, \ldots p$, see Subsection 5.4.4) evaluated at a vector $\delta\bar{\sigma}$ can be efficiently computed without saving the Jacobian $J(\bar{\sigma})$ of the residual function.

Theorem 5.5.2 *Let* $\delta\bar{\sigma} = (\delta\bar{\sigma}_0, \ldots, \delta\bar{\sigma}_{M-1}) \in I\!R^n$ *with* $\delta\bar{\sigma}_j = (\delta\bar{\sigma}_{1,j}, \ldots, \delta\bar{\sigma}_{N-1,j}) \in I\!R^{(N-1)}$, $j = 0, \ldots, M-1$, $n = M\,(N-1)$, *and set*

$$s_j := \sum_{k=0}^{j-1} \tfrac{\partial}{\partial\bar{\sigma}_k} z_j(\bar{\sigma})\,\delta\bar{\sigma}_k\,. \tag{5.5.2}$$

Then

$$J(\bar{\sigma})\delta\bar{\sigma} = \bar{s} = (\bar{s}_{j_1}, \bar{s}_{j_2}, \ldots, \bar{s}_{j_p}) \in I\!R^m\,, \qquad \bar{s}_{j_i} = A_{j_i} s_{j_i}\,, \quad i = 1, \ldots, p\,, \tag{5.5.3}$$

where $j_i \in I$ *with* I *defined in (5.4.60) and* s_{j+1}, $j = 1, \ldots, M-1$, *can be computed recursively by solving*

$$U_j(\bar{\sigma})\,s_{j+1} = V_j(\bar{\sigma})\,s_j + F_j(\bar{\sigma})\,, \tag{5.5.4}$$

with s_1 *is the solution to*

$$U_0(\bar{\sigma})\,s_1 = F_0(\bar{\sigma}) \tag{5.5.5}$$

and $F_j(\bar{\sigma})$, $j = 0, \ldots, M-1$, *are given by*

$$F_j(\bar{\sigma}) = \alpha\,\Sigma_j\,\delta\Sigma_j\,(H\,((1-\theta)z_j(\bar{\sigma}) + \theta z_{j+1}(\bar{\sigma})) + h_j(\bar{\sigma})) \tag{5.5.6}$$

with

$$\begin{aligned}
\Sigma_j &= \text{diag}(\bar{\sigma}_{1,j}, \ldots, \bar{\sigma}_{N-1,j})\,,\\
\delta\Sigma_j &= \text{diag}(\delta\bar{\sigma}_{1,j}, \ldots, \delta\bar{\sigma}_{N-1,j})\,,\\
h_j(\bar{\sigma}) &= ((1+\tfrac{\Delta x}{2})((1-\theta)z_{0,j}(\bar{\sigma}) + \theta z_{0,j+1}(\bar{\sigma})), 0, \ldots, 0)^T\,,\\
H &= \begin{bmatrix} -2 & (1-\frac{\Delta x}{2}) & & & \\ (1+\frac{\Delta x}{2}) & -2 & (1-\frac{\Delta x}{2}) & & \\ & \ddots & \ddots & \ddots & \\ & & (1+\frac{\Delta x}{2}) & -2 & (1-\frac{\Delta x}{2}) \\ & & & (1+\frac{\Delta x}{2}) & -2 \end{bmatrix},
\end{aligned} \tag{5.5.7}$$

and $z_{0,j}(\bar{\sigma})$, $j = 0, \ldots, M$, *defined in (5.4.25).*

Proof: If we set $A = \text{diag}(A_{j_1}, A_{j_2}, \ldots, A_{j_p})$, i.e. A is a block diagonal matrix, then

$$R(\bar{\sigma}) = \begin{pmatrix} A_{j_1}\,z_{j_1}(\bar{\sigma}) - \hat{C}_{j_1} \\ A_{j_2}\,z_{j_2}(\bar{\sigma}) - \hat{C}_{j_2} \\ \vdots \\ A_{j_p}\,z_{j_p}(\bar{\sigma}) - \hat{C}_{j_p} \end{pmatrix} = A\,\hat{z}(\bar{\sigma}) - c_{observ}$$

with $\hat{z}(\bar{\sigma}) = (z_{j_1}(\bar{\sigma}), z_{j_2}(\bar{\sigma}), \ldots, z_{j_p}(\bar{\sigma}))^T$ and $c_{observ} = (\hat{C}_{j_1}, \hat{C}_{j_2}, \ldots, \hat{C}_{j_p})^T$. Note that A_{j_i} and $\hat{C}_{j_i}$, $i = 1, \ldots, p$, do not depend on $\bar{\sigma}$. Therefore,

$$J(\bar{\sigma})\delta\bar{\sigma} = A\, \tfrac{d}{d\bar{\sigma}} \hat{z}(\bar{\sigma})\delta\bar{\sigma}\,.$$

Note that $z_j(\bar{\sigma})$, $j = 1, \ldots, M$, is defined recursively by (5.4.28). Since $\tilde{D}_j(\bar{\sigma})$, $\tilde{B}_j(\bar{\sigma})$ (see (5.4.19), (5.4.23), (5.4.29) and (5.4.30)) and hence $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$ are only dependend on $\bar{\sigma}_j$ it follows that $z_j(\bar{\sigma})$ depends only on $\bar{\sigma}_0, \bar{\sigma}_1, \ldots, \bar{\sigma}_{j-1}$ and therefore,

$$\tfrac{\partial}{\partial\bar{\sigma}_k} z_j(\bar{\sigma})\, \delta\bar{\sigma}_k = 0\,, \qquad k = j, \ldots, M-1\,.$$

Hence, we get

$$J(\bar{\sigma})\delta\bar{\sigma} = A\, \tfrac{d}{d\bar{\sigma}} \hat{z}(\bar{\sigma})\delta\bar{\sigma} = A\, s = (A_{j_1}s_{j_1}, A_{j_2}s_{j_2}, \ldots, A_{j_p}s_{j_p})^T = \bar{s}$$

where

$$\begin{aligned} s &= (s_{j_1}, s_{j_2}, \ldots, s_{j_p})^T\,, \\ s_{j_i} &= \sum_{k=0}^{j_i-1} \tfrac{\partial}{\partial\bar{\sigma}_k} z_{j_i}(\bar{\sigma})\, \delta\bar{\sigma}_k\,, \quad \text{and} \\ \bar{s}_{j_i} &= A_{j_i}s_{j_i}\,. \end{aligned}$$

Furthermore,

$$\begin{aligned} \tfrac{\partial}{\partial\bar{\sigma}_k} \tilde{D}^j_l(\bar{\sigma})\, \delta\bar{\sigma}_k &= 0\,, \quad k \neq j\,, \quad k = 0, \ldots, M-1\,, \\ \tfrac{\partial}{\partial\bar{\sigma}_k} \tilde{B}_j(\bar{\sigma})\, \delta\bar{\sigma}_k &= 0\,, \quad k \neq j\,, \quad k = 0, \ldots, M-1\,, \end{aligned} \tag{5.5.8}$$

$l = 1, \ldots, N-1$, $j = 0, \ldots, M-1$, such that also

$$\begin{aligned} \tfrac{\partial}{\partial\bar{\sigma}_k} U^j_l(\bar{\sigma})\, \delta\bar{\sigma}_k &= 0\,, \quad k \neq j\,, \quad k = 0, \ldots, M-1\,, \\ \tfrac{\partial}{\partial\bar{\sigma}_k} V^j_l(\bar{\sigma})\, \delta\bar{\sigma}_k &= 0\,, \quad k \neq j\,, \quad k = 0, \ldots, M-1\,. \end{aligned} \tag{5.5.9}$$

Taking the partial derivative with respect to $\bar{\sigma}_k$ of the right and left hand side of (5.4.28), i.e.

$$\tfrac{\partial}{\partial\bar{\sigma}_k} \left(U_j(\bar{\sigma})\; z_{j+1}(\bar{\sigma})\right) \delta\bar{\sigma}_k \;=\; \tfrac{\partial}{\partial\bar{\sigma}_k} \Big(V_j(\bar{\sigma})\; z_j(\bar{\sigma}) \;+\; \tilde{B}_j(\bar{\sigma})\Big)\, \delta\bar{\sigma}_k$$

and using Lemma 5.5.1 we get

$$\begin{aligned} & U_j(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} z_{j+1}(\bar{\sigma})\delta\bar{\sigma}_k + \sum_{l=1}^{N-1} z_{l,j+1}(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} U^j_l(\bar{\sigma})\delta\bar{\sigma}_k \\ =\;& V_j(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} z_j(\bar{\sigma})\delta\bar{\sigma}_k + \sum_{l=1}^{N-1} z_{l,j}(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} V^j_l(\bar{\sigma})\delta\bar{\sigma}_k + \tfrac{\partial}{\partial\bar{\sigma}_k} \tilde{B}_j(\bar{\sigma})\delta\bar{\sigma}_k\,, \end{aligned} \tag{5.5.10}$$

$j, k = 0, \ldots, M-1$. A rearrangement of the terms in (5.5.10) and (5.5.8) and (5.5.9) yields for $k < j$

$$U_j(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} z_{j+1}(\bar{\sigma})\delta\bar{\sigma}_k = V_j(\bar{\sigma}) \tfrac{\partial}{\partial\bar{\sigma}_k} z_j(\bar{\sigma})\delta\bar{\sigma}_k\,, \tag{5.5.11}$$

for $k = j$

$$\begin{aligned} U_j(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j} z_{j+1}(\bar\sigma)\delta\bar\sigma_j &= V_j(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j} z_j(\bar\sigma)\delta\bar\sigma_j + \tfrac{\partial}{\partial\bar\sigma_j}\tilde B_j(\bar\sigma)\delta\bar\sigma_j + \\ &\quad \sum_{l=1}^{N-1}\Big(z_{l,j}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}V_l^j(\bar\sigma) - z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}U_l^j(\bar\sigma)\Big)\,\delta\bar\sigma_j \\ &= \tfrac{\partial}{\partial\bar\sigma_j}\tilde B_j(\bar\sigma)\delta\bar\sigma_j + \\ &\quad \sum_{l=1}^{N-1}\Big(z_{l,j}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}V_l^j(\bar\sigma) - z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}U_l^j(\bar\sigma)\Big)\,\delta\bar\sigma_j\,, \end{aligned} \tag{5.5.12}$$

($\tfrac{\partial}{\partial\bar\sigma_j} z_j(\bar\sigma)\delta\bar\sigma_j = 0$) and for $k > j$

$$U_j(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_k} z_{j+1}(\bar\sigma)\delta\bar\sigma_k = 0\,. \tag{5.5.13}$$

Next, we examine the terms in the sum on the very right hand side of Equation 5.5.12. For the partial derivative of the lth column of the matrices $U_j(\bar\sigma)$ and $V_j(\bar\sigma)$ with respect to $\bar\sigma_j$, we get

$$\begin{aligned} \tfrac{\partial}{\partial\bar\sigma_j}U_l^j(\bar\sigma) &= \tfrac{\partial}{\partial\bar\sigma_j}\Big(e_l + \theta\,\tilde D_l^j(\bar\sigma)\Big) &&= \theta\,\tfrac{\partial}{\partial\bar\sigma_j}\tilde D_l^j(\bar\sigma)\,, \\ \tfrac{\partial}{\partial\bar\sigma_j}V_l^j(\bar\sigma) &= \tfrac{\partial}{\partial\bar\sigma_j}\Big(e_l - (1-\theta)\,\tilde D_l^j(\bar\sigma)\Big) &&= -(1-\theta)\,\tfrac{\partial}{\partial\bar\sigma_j}\tilde D_l^j(\bar\sigma)\,. \end{aligned}$$

Furthermore, since

$$\tfrac{\partial}{\partial\bar\sigma_{i,j}}u_l^j(\bar\sigma)\delta\bar\sigma_{i,j} = \tfrac{\partial}{\partial\bar\sigma_{i,j}}m_l^j(\bar\sigma)\delta\bar\sigma_{i,j} = \tfrac{\partial}{\partial\bar\sigma_{i,j}}v_l^j(\bar\sigma)\delta\bar\sigma_{i,j} = 0 \quad \text{for} \quad i \neq l$$

and

$$\begin{aligned} \tfrac{\partial}{\partial\bar\sigma_{l,j}}v_l^j(\bar\sigma) &= -\left(1+\tfrac{\Delta x}{2}\right)\bar\sigma_{l,j} \\ \tfrac{\partial}{\partial\bar\sigma_{l,j}}m_l^j(\bar\sigma) &= 2\,\bar\sigma_{l,j} \\ \tfrac{\partial}{\partial\bar\sigma_{l,j}}u_l^j(\bar\sigma) &= -\left(1-\tfrac{\Delta x}{2}\right)\bar\sigma_{l,j} \end{aligned}$$

the Jacobian of the lth column of $\tilde D_j(\bar\sigma)$, i.e. $\tfrac{\partial}{\partial\bar\sigma_j}\tilde D_l^j(\bar\sigma)$, is a diagonal matrix that is given by

$$\begin{aligned} \tfrac{\partial}{\partial\bar\sigma_j}\tilde D_l^j(\bar\sigma) &= \alpha\ \text{diag}\Big(0,\dots,0,\tfrac{\partial}{\partial\bar\sigma_{l-1,j}}u_{l-1}^j(\bar\sigma),\tfrac{\partial}{\partial\bar\sigma_{l,j}}m_l^j(\bar\sigma),\tfrac{\partial}{\partial\bar\sigma_{l+1,j}}v_{l+1}^j(\bar\sigma),0,\dots,0\Big) \\ &= \alpha\ \text{diag}\left(0,\dots,0,-\left(1-\tfrac{\Delta x}{2}\right)\bar\sigma_{l-1,j},2\,\bar\sigma_{l,j},-\left(1+\tfrac{\Delta x}{2}\right)\bar\sigma_{l+1,j},0,\dots,0\right) \end{aligned}$$

for $l = 2, \ldots, N-2$ and for $l = 1$ and $l = N-1$ the Jacobian is given by

$$\begin{aligned}
\tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_1^j(\bar{\sigma}) &= \alpha \; \text{diag} \left(\tfrac{\partial}{\partial \bar{\sigma}_{1,j}} m_1^j(\bar{\sigma}) \, , \tfrac{\partial}{\partial \bar{\sigma}_{2,j}} v_2^j(\bar{\sigma}) \, , 0, \ldots, 0 \right) \\
&= \alpha \; \text{diag} \left(2 \, \bar{\sigma}_{1,j} \, , - \left(1 + \tfrac{\Delta x}{2} \right) \bar{\sigma}_{2,j} \, , 0, \ldots, 0 \right) \\
\tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_{N-1}^j(\bar{\sigma}) \delta \bar{\sigma}_j &= \alpha \; \text{diag} \left(0, \ldots, 0, \tfrac{\partial}{\partial \bar{\sigma}_{N-2,j}} u_{N-2}^j(\bar{\sigma}) \, , \tfrac{\partial}{\partial \bar{\sigma}_{N-1,j}} m_{N-1}^j(\bar{\sigma}) \right) \\
&= \alpha \; \text{diag} \left(0, \ldots, 0, - \left(1 - \tfrac{\Delta x}{2} \right) \bar{\sigma}_{N-2,j} \, , 2 \, \bar{\sigma}_{N-1,j} \right) \; .
\end{aligned}$$

Thus,

$$\begin{aligned}
z_{l,j}(\bar{\sigma}) \, \tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_l^j(\bar{\sigma}) \delta \bar{\sigma}_j \;=\; \alpha \; \Big(& 0, \ldots, 0, - \left(1 - \tfrac{\Delta x}{2} \right) \bar{\sigma}_{l-1,j} \, \delta \bar{\sigma}_{l-1,j} \, z_{l,j}(\bar{\sigma}), 2 \, \bar{\sigma}_{l,j} \, \delta \bar{\sigma}_{l,j} \, z_{l,j}(\bar{\sigma}), \\
& - \left(1 + \tfrac{\Delta x}{2} \right) \bar{\sigma}_{l+1,j} \, \delta \bar{\sigma}_{l+1,j} \, z_{l,j}(\bar{\sigma}), 0, \ldots, 0 \Big)^T
\end{aligned}$$

for $l = 2, \ldots, N-2$ and

$$\begin{aligned}
z_{1,j}(\bar{\sigma}) \, \tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_1^j(\bar{\sigma}) \delta \bar{\sigma}_j &= \alpha \; \big(2 \, \bar{\sigma}_{1,j} \, \delta \bar{\sigma}_{1,j} \, z_{1,j}(\bar{\sigma}) \, , \\
& \qquad - \left(1 + \tfrac{\Delta x}{2} \right) \bar{\sigma}_{2,j} \, \delta \bar{\sigma}_{2,j} \, z_{1,j}(\bar{\sigma}) \, , 0, \ldots, 0 \big) \\
z_{N-1,j}(\bar{\sigma}) \, \tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_{N-1}^j(\bar{\sigma}) \delta \bar{\sigma}_j &= \alpha \; \big(0, \ldots, 0, - \left(1 - \tfrac{\Delta x}{2} \right) \bar{\sigma}_{N-2,j} \, \delta \bar{\sigma}_{N-2,j} \, z_{N-1,j}(\bar{\sigma}) \, , \\
& \qquad 2 \, \bar{\sigma}_{N-1,j} \, \delta \bar{\sigma}_{N-1,j} \, z_{N-1,j}(\bar{\sigma}))
\end{aligned}$$

such that

$$\begin{aligned}
\sum_{l=1}^{N-1} z_{l,j}(\bar{\sigma}) \tfrac{\partial}{\partial \bar{\sigma}_j} V_l^j(\bar{\sigma}) \delta \bar{\sigma}_j \;&=\; \sum_{l=1}^{N-1} z_{l,j}(\bar{\sigma}) \tfrac{\partial}{\partial \bar{\sigma}_j} \left(e_l - (1 - \theta) \, \tilde{D}_l^j(\bar{\sigma}) \right) \, \delta \bar{\sigma}_j \\
= \; - (1 - \theta) \sum_{l=1}^{N-1} & z_{l,j}(\bar{\sigma}) \tfrac{\partial}{\partial \bar{\sigma}_j} \tilde{D}_l^j(\bar{\sigma}) \delta \bar{\sigma}_j \\
= \; - (1 - \theta) \, \alpha & \begin{pmatrix}
\left(2 \, z_{1,j}(\bar{\sigma}) - (1 - \tfrac{\Delta x}{2}) \, z_{2,j}(\bar{\sigma}) \right) \, \bar{\sigma}_{1,j} \, \delta \bar{\sigma}_{1,j} \\
\vdots \\
\left(-(1 + \tfrac{\Delta x}{2}) \, z_{l-1,j}(\bar{\sigma}) + 2 \, z_{l,j}(\bar{\sigma}) - (1 - \tfrac{\Delta x}{2}) \, z_{l+1,j}(\bar{\sigma}) \right) \, \bar{\sigma}_{l,j} \, \delta \bar{\sigma}_{l,j} \\
\vdots \\
\left(-(1 + \tfrac{\Delta x}{2}) \, z_{N-2,j}(\bar{\sigma}) + 2 \, z_{N-1,j}(\bar{\sigma}) \right) \, \bar{\sigma}_{N-1,j} \, \delta \bar{\sigma}_{N-1,j}
\end{pmatrix}
\end{aligned}$$

$$= (1-\theta)\,\alpha\,\Sigma_j\,\delta\Sigma_j \begin{pmatrix} -2\,z_{1,j}(\bar\sigma) + (1-\frac{\Delta x}{2})\,z_{2,j}(\bar\sigma) \\ \vdots \\ (1+\frac{\Delta x}{2})\,z_{l-1,j}(\bar\sigma) - 2\,z_{l,j}(\bar\sigma) + (1-\frac{\Delta x}{2})\,z_{l+1,j}(\bar\sigma) \\ \vdots \\ (1+\frac{\Delta x}{2})\,z_{N-2,j}(\bar\sigma) - 2\,z_{N-1,j}(\bar\sigma) \end{pmatrix}$$

$$= (1-\theta)\,\alpha\,\Sigma_j\,\delta\Sigma_j\,H\,z_j(\bar\sigma)$$

i.e.

$$\sum_{l=1}^{N-1} z_{l,j}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}V_l^j(\bar\sigma)\delta\bar\sigma_j \;=\; (1-\theta)\,\alpha\,\Sigma_j\,\delta\Sigma_j\,H\,z_j(\bar\sigma) \tag{5.5.14}$$

with Σ_j, $\delta\Sigma_j$ and H defined in (5.5.7). Similarly, it follows that

$$\begin{aligned} \sum_{l=1}^{N-1} z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}U_l^j(\bar\sigma)\delta\bar\sigma_j &= \sum_{l=1}^{N-1} z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}\left(e_l + \theta\,\tilde D_l^j(\bar\sigma)\right)\delta\bar\sigma_j \\ &= \theta\sum_{l=1}^{N-1} z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}\tilde D_l^j(\bar\sigma)\delta\bar\sigma_j \\ &= -\,\theta\,\alpha\,\Sigma_j\,\delta\Sigma_j\,H\,z_{j+1}(\bar\sigma)\,. \end{aligned} \tag{5.5.15}$$

Furthermore,

$$\begin{aligned} &\tfrac{\partial}{\partial\bar\sigma_j}\tilde B_j(\bar\sigma)\delta\bar\sigma_j \\ &\quad= -\alpha\,\left(-(1+\tfrac{\Delta x}{2})\bar\sigma_{1,j}\,\left((1-\theta)\,z_{0,j}(\bar\sigma) + \theta z_{0,j+1}(\bar\sigma)\right)\delta\bar\sigma_{1,j}, 0, \ldots, 0\right)^T \\ &\quad= \alpha\,\Sigma_j\delta\Sigma_j\,h_j(\bar\sigma) \end{aligned} \tag{5.5.16}$$

with $h_j(\bar\sigma)$ defined in (5.5.7). Note that the values at the boundary, i.e. $z_{0,j}(\bar\sigma)\,, z_{N,j}(\bar\sigma)$, do not depend on the volatility parameter $\bar\sigma$. Using (5.5.14), (5.5.15) and (5.5.16), we get

$$\begin{aligned} &\sum_{l=1}^{N-1}\left(z_{l,j}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}V_l^j(\bar\sigma) - z_{l,j+1}(\bar\sigma)\tfrac{\partial}{\partial\bar\sigma_j}U_l^j(\bar\sigma)\right)\delta\bar\sigma_j + \tfrac{\partial}{\partial\bar\sigma_j}\tilde B_j(\bar\sigma)\delta\bar\sigma_j \\ &\qquad= \alpha\,\Sigma_j\,\delta\Sigma_j\,\left(H\left((1-\theta)z_j(\bar\sigma) + \theta z_{j+1}(\bar\sigma)\right) + h_j(\bar\sigma)\right)\,. \end{aligned} \tag{5.5.17}$$

Now, we show by induction over j that s_j, $j = 1, \ldots, M$, can be computed recursively by using (5.5.4). We start with the computation of $s_1 = \frac{\partial}{\partial\bar\sigma_0}z_1(\bar\sigma)\,\delta\bar\sigma_0$. The approximations $z_1(\bar\sigma)$ of the value of the European call option in the time layer ΔT are given by the solution of the linear system of equations

$$U_0(\bar\sigma)\,z_1(\bar\sigma) \;=\; V_0(\bar\sigma)\,z_0(\bar\sigma) \;+\; \tilde B_0(\bar\sigma)\,.$$

The partial derivative with respect to $\bar{\sigma}_0$ of both sides of the previous equation are given by

$$\begin{aligned} U_0(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_0}z_1(\bar{\sigma})\delta\bar{\sigma}_0 &= \tfrac{\partial}{\partial\bar{\sigma}_0}\tilde{B}_0(\bar{\sigma})\delta\bar{\sigma}_0 + \sum_{l=1}^{N-1}\Big(z_{l,0}(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_0}V_l^0(\bar{\sigma}) - z_{l,1}(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_0}U_l^0(\bar{\sigma})\Big)\delta\bar{\sigma}_0 \\ &= \alpha\,\Sigma_0\,\delta\Sigma_0\,\left(H((1-\theta)z_0(\bar{\sigma})+\theta z_1(\bar{\sigma}))+h_0(\bar{\sigma})\right) \\ &= F_0(\bar{\sigma})\,. \end{aligned}$$

where we have used Lemma 5.5.1 together with the transformations (5.5.10) and (5.5.12) for $j = k = 0$ and (5.5.17). Since $s_1 = \frac{\partial}{\partial\bar{\sigma}_0}z_1(\bar{\sigma})\delta\bar{\sigma}_0$ we have proven (5.5.5).

Next, we prove the induction step from j to $j+1$. Using the identity (5.5.2) of s_{j+1}, we get

$$U_j(\bar{\sigma})s_{j+1} = U_j(\bar{\sigma})\left(\sum_{k=0}^{j}\tfrac{\partial}{\partial\bar{\sigma}_k}z_{j+1}(\bar{\sigma})\,\delta\bar{\sigma}_k\right) = \sum_{k=0}^{j}U_j(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_k}z_{j+1}(\bar{\sigma})\,\delta\bar{\sigma}_k$$

and with (5.5.11), (5.5.12), (5.5.13), (5.5.17) and the identity (5.5.2) for s_j it follows

$$\begin{aligned} U_j(\bar{\sigma})s_{j+1} &= \sum_{k=0}^{j} U_j(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_k}z_{j+1}(\bar{\sigma})\,\delta\bar{\sigma}_k \\ &= \sum_{k=0}^{j-1}\Big(V_j(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_k}z_j(\bar{\sigma})\delta\bar{\sigma}_k\Big) + \\ &\qquad \tfrac{\partial}{\partial\bar{\sigma}_j}\tilde{B}_j(\bar{\sigma})\delta\bar{\sigma}_j + \sum_{l=1}^{N-1}\Big(z_{l,j}(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_j}V_l^j(\bar{\sigma}) - z_{l,j+1}(\bar{\sigma})\tfrac{\partial}{\partial\bar{\sigma}_j}U_l^j(\bar{\sigma})\Big)\delta\bar{\sigma}_j \\ &= V_j(\bar{\sigma})\left(\sum_{k=0}^{j-1}\tfrac{\partial}{\partial\bar{\sigma}_k}z_j(\bar{\sigma})\delta\bar{\sigma}_k\right) + \\ &\qquad \alpha\,\Sigma_j\,\delta\Sigma_j\,\left(H\left((1-\theta)z_j(\bar{\sigma})+\theta z_{j+1}(\bar{\sigma})\right)+h_j(\bar{\sigma})\right) \\ &= V_j(\bar{\sigma})\,s_j + F_j(\bar{\sigma}) \end{aligned}$$

which confirms (5.5.4). ■

The computational effort for one evaluation of $J(\bar{\sigma})\delta\bar{\sigma}$ is composed of the effort necessary for setting up the system of linear equations (5.5.5) and (5.5.4) and the solution of it. The coefficients of the matrices $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$ are composed of the sum of the particular elements of the matrix $\tilde{D}_j(\bar{\sigma})$ multiplied by $-(1-\theta)$ and θ, respectively, and the corresponding elements of the identity matrix. Thus, the number of multiplications necessary to compute each of the three diagonals of $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$ consists of the computational effort for the determination of the elements of $\tilde{D}_j(\bar{\sigma})$ and the subsequent multiplication with $-(1-\theta)$ and θ, respectively, which is given in the relevant case by

computation of $\tilde{D}_j(\bar{\sigma})$: $6\,(N-2)+N = 7\,N-12$ multiplications,
computation of $U_j(\bar{\sigma})$: $3\,N-5$ multiplications,
computation of $V_j(\bar{\sigma})$: $3\,N-5$ multiplications.

The right hand side of (5.5.5) consists of the vector $F_0(\bar{\sigma})$ and that of (5.5.4) is composed of the matrix vector product $V_j(\bar{\sigma})\, s_j$ and $F_j(\bar{\sigma})$, $j = 1, \ldots, M-1$. Since $V_j(\bar{\sigma})$ is a tridiagonal matrix the effort for the computation of the matrix–vector product amounts to $3\,N-5$ multiplications. The vector $F_j(\bar{\sigma})$ is composed of the sum of a matrix–vector product and a vector that is multiplied by two diagonal matrices from the left and a scalar. The effort for the matrix–vector product is again $3\,N-5$ multiplications since the matrix H is also tridiagonal. For the computation of the vector $(1-\theta)z_j(\bar{\sigma}) + \theta z_{j+1}(\bar{\sigma})$ are $2\,(N-1)$ multiplications necessary. The evaluation of $h_j(\bar{\sigma})$ consists only in the calculation of the first element which costs three multiplications. Thus, the effort for the computation of $F_j(\bar{\sigma})$ amounts to $3\,N-3+3\,N-5+2\,N-2+3 = 8\,N-7$ multiplications for $j = 0, \ldots, M-1$. Finally, for the solution of the linear tridiagonal systems (5.5.5) and (5.5.4) for $j = 0, \ldots, M-1$ are $5\,(N-1)-4$ multiplications necessary when using a LU factorization (see for instance Schwarz [94]) such that the total computational effort for one evaluation of the matrix–vector product $J(\bar{\sigma})\delta\bar{\sigma}$ adds up to

$$\begin{aligned} & M \cdot (5\,N-9+8\,N-7+7\,N-12+3\,N-5) + (M-1)\,(3\,N-5+3\,N-5) \\ & \quad = \; M \; (29\,N-43) - (6\,N-10) \quad \text{multiplications}\,. \end{aligned} \tag{5.5.18}$$

Thus, the computational effort for one evaluation of $J(\bar{\sigma})\delta\bar{\sigma}$ is of order $\mathcal{O}(MN)$.

In the next theorem, we show how $J(\bar{\sigma})^T\delta c$ can be efficiently evaluated also without storing the Jacobian $J(\bar{\sigma})$.

Theorem 5.5.3 *Let* $\bar{\sigma} = (\bar{\sigma}_0, \ldots, \bar{\sigma}_{M-1}) \in I\!R^{M(N-1)}$ *and* $\delta c = (\delta c_1, \delta c_2, \ldots, \delta c_p) \in I\!R^m$ *with* $\delta c_k \in I\!R^{m_{j_k}}$, $j_k \in I$, $k = 1, \ldots, p$, *and* $1 \le j_1 < j_2 < \ldots < j_p = M$. *Set for* $k = 1, \ldots, p$

$$\begin{aligned} \mu_k^{j_k} &= A_{j_k}^T \delta c_k \in I\!R^{N-1} \\ \mu_k^j &= V_j(\bar{\sigma})^T \zeta_k^j \in I\!R^{N-1}\,, \quad j = j_k-1, j_k-2, \ldots, 0\,, \end{aligned} \tag{5.5.19}$$

where ζ_k^j, $j = j_k-1, j_k-2, \ldots, 0$, $k = 1, \ldots, p$, *is the solution to*

$$U_j(\bar{\sigma})^T \zeta_k^j = \mu_k^{j+1}\,. \tag{5.5.20}$$

Then

$$J(\bar{\sigma})^T \delta c = \tilde{\sigma} = (\tilde{\sigma}_0, \ldots, \tilde{\sigma}_{M-1}) \in I\!R^{M(N-1)}\,, \tag{5.5.21}$$

$\tilde{\sigma}_j \in I\!R^{N-1}$, $j = 0, \ldots, M-1$, *can be computed by*

$$\tilde{\sigma}_j = \Lambda_j\, \Sigma_j \sum_{z=k}^{p} \zeta_z^j\,, \quad j_{k-1} \le j \le j_k - 1\,, \quad k = 1, \ldots, p\,, \tag{5.5.22}$$

with $j_0 = 0$ *and*

$$\Lambda_j = \text{diag}(\lambda_{1,j}, \ldots, \lambda_{N-1,j}) \tag{5.5.23}$$

where $\lambda_{i,j}$, $i = 1, \ldots, N-1$, *is the* i*th component of*

$$\lambda_j = \alpha\; (H\,((1-\theta)\, z_j(\bar{\sigma}) + \theta\, z_{j+1}(\bar{\sigma})) + h_j(\bar{\sigma})) \tag{5.5.24}$$

and with H *and* $h_j(\bar{\sigma})$, $j = 0, \ldots, M-1$, *defined in* (5.5.7).

Proof: For arbitrary $\delta\bar{\sigma} = (\delta\bar{\sigma}_0, \ldots, \delta\bar{\sigma}_{M-1}) \in I\!R^{M(N-1)}$, $\delta\bar{\sigma}_l \in I\!R^{N-1}\,, l = 0, \ldots, M-1$, we have according to Theorem 5.5.2 and (5.5.21), (5.5.3) and (5.5.19),

$$\tilde{\sigma}^T \delta\bar{\sigma} = \delta c^T J(\bar{\sigma})\delta\bar{\sigma} = \delta c^T \bar{s} = \sum_{k=1}^{p} \delta c_k^T \bar{s}_{j_k} = \sum_{k=1}^{p} \delta c_k^T A_{j_k} s_{j_k} = \sum_{k=1}^{p} (\mu_k^{j_k})^T s_{j_k}\,. \tag{5.5.25}$$

For $j = j_k, j_k - 1, \ldots, 1$ and $k = 1, \ldots, p$ we obtain with (5.5.4), (5.5.20), (5.5.19), (5.5.5) and Theorem 5.5.2

$$\begin{aligned}
(\mu_k^j)^T s_j &= (\mu_k^j)^T \left(U_{j-1}(\bar{\sigma})^{-1} V_{j-1}(\bar{\sigma})\, s_{j-1} + U_{j-1}(\bar{\sigma})^{-1} F_{j-1}(\bar{\sigma})\right) \\
&= (\mu_k^j)^T U_{j-1}(\bar{\sigma})^{-1} V_{j-1}(\bar{\sigma})\, s_{j-1} + (U_{j-1}(\bar{\sigma})^{-T} \mu_k^j)^T F_{j-1}(\bar{\sigma}) \\
&= (V_{j-1}(\bar{\sigma})^T U_{j-1}(\bar{\sigma})^{-T} \mu_k^j)^T s_{j-1} + (\zeta_k^{j-1})^T F_{j-1}(\bar{\sigma}) \\
&= (V_{j-1}(\bar{\sigma})^T \zeta_k^{j-1})^T s_{j-1} + (\zeta_k^{j-1})^T F_{j-1}(\bar{\sigma}) \\
&= (\mu_k^{j-1})^T s_{j-1} + (\zeta_k^{j-1})^T F_{j-1}(\bar{\sigma}) \\
(\mu_k^1)^T s_1 &= (\mu_k^1)^T U_0(\bar{\sigma})^{-1} F_0(\bar{\sigma}) \\
&= (U_0(\bar{\sigma})^{-T} \mu_k^1)^T F_0(\bar{\sigma}) \\
&= (\zeta_k^0)^T F_0(\bar{\sigma})
\end{aligned} \tag{5.5.26}$$

where $F_j(\bar{\sigma})$, $j = 0, \ldots, M-1$, is defined by (see (5.5.6))

$$F_j(\bar{\sigma}) = \alpha\, \Sigma_j\, \delta\Sigma_j\, \left(H\left((1-\theta) z_j(\bar{\sigma}) + \theta z_{j+1}(\bar{\sigma})\right) + h_j(\bar{\sigma})\right) = \Sigma_j\, \delta\Sigma_j\, \lambda_j$$

with λ_j defined in (5.5.24). Using (5.5.26) and substituting recursively into (5.5.25) yields

$$\begin{aligned}
\tilde{\sigma}^T \delta\bar{\sigma} &= \sum_{k=1}^{p} (\mu_k^{j_k})^T s_{j_k} \\
&= \sum_{k=1}^{p} \sum_{j=1}^{j_k} (\zeta_k^{j-1})^T F_{j-1}(\bar{\sigma}) \\
&= \sum_{k=1}^{p} \sum_{j=1}^{j_k} (\zeta_k^{j-1})^T \Sigma_{j-1}\, \delta\Sigma_{j-1}\, \lambda_{j-1} \\
&= \sum_{k=1}^{p} \sum_{j=j_{k-1}+1}^{j_k} \sum_{z=k}^{p} (\zeta_z^{j-1})^T \Sigma_{j-1}\, \delta\Sigma_{j-1}\, \lambda_{j-1} \\
&= \sum_{k=1}^{p} \sum_{j=j_{k-1}}^{j_k - 1} \sum_{z=k}^{p} (\Sigma_j\, \zeta_z^j)^T \delta\Sigma_j\, \lambda_j
\end{aligned}$$

The transformations from the third row to the fourth row of this equation consists in a rearrangement of the terms in the sum. For elucidation, we have stated exemplary the different terms of this sum

		j							
		1	2	3	4	5	6	7	8
			j_1	j_2		j_3	j_4		j_5
k	1	$(\zeta_1^0)^T F_0$	$(\zeta_1^1)^T F_1$						
	2	$(\zeta_2^0)^T F_0$	$(\zeta_2^1)^T F_1$	$(\zeta_2^2)^T F_2$					
	3	$(\zeta_3^0)^T F_0$	$(\zeta_3^1)^T F_1$	$(\zeta_3^2)^T F_2$	$(\zeta_3^3)^T F_3$	$(\zeta_3^4)^T F_4$			
	4	$(\zeta_4^0)^T F_0$	$(\zeta_4^1)^T F_1$	$(\zeta_4^2)^T F_2$	$(\zeta_4^3)^T F_3$	$(\zeta_4^4)^T F_4$	$(\zeta_4^5)^T F_5$		
	5	$(\zeta_5^0)^T F_0$	$(\zeta_5^1)^T F_1$	$(\zeta_5^2)^T F_2$	$(\zeta_5^3)^T F_3$	$(\zeta_5^4)^T F_4$	$(\zeta_5^5)^T F_5$	$(\zeta_5^6)^T F_6$	$(\zeta_5^7)^T F_7$

TABLE 5.1: Exemplary listing of the terms in the sum of $\tilde{\sigma}^T \delta\bar{\sigma}$ for $p = 5, j_1 = 2\,, j_2 = 3\,, j_3 = 5, j_4 = 6, j_5 = 8$

in Table 5.1. In the third row of this equation, the terms are added up row for row with regard to Table 5.1. In contrast, in the fourth row of the equation, the different terms are summed up column for column. Furthermore, we have used that Σ_j and $\delta\Sigma_j$ are diagonal matrices, i.e. $\Sigma_j = (\Sigma_j)^T$ and $\delta\Sigma_j = (\delta\Sigma_j)^T$.

Then, for $k = 1, \ldots, p$ and $j = j_{k-1}, \ldots, j_k - 1$ we get with $u^T v = \text{tr}[uv^T]$, $u, v \in I\!R^{N-1}$

$$\begin{aligned}
\tilde{\sigma}_j^T \delta\bar{\sigma}_j &= \sum_{z=k}^{p} (\Sigma_j\, \zeta_z^j)^T\, \delta\Sigma_j\, \lambda_j \\
&= \sum_{z=k}^{p} \text{tr}\left[\Sigma_j\, \zeta_z^j\, (\delta\Sigma_j\, \lambda_j)^T\right] \\
&= \sum_{z=k}^{p} \text{tr}\left[\Sigma_j\, \zeta_z^j\, \lambda_j^T\, \delta\Sigma_j\right] \\
&= \sum_{z=k}^{p} \text{tr}\left[\lambda_j\, (\Sigma_j\, \zeta_z^j)^T\, \delta\Sigma_j\right] \\
&= \sum_{z=k}^{p} \text{tr}\left[\left(\sum_{i=1}^{N-1} \lambda_{q,j}\, (\Sigma_j\, \zeta_z^j)_i\, (\delta\Sigma_j)_{i,s}\right)_{1\leq q,s\leq N-1}\right] \\
&= \sum_{z=k}^{p} \sum_{q=1}^{N-1} \sum_{i=1}^{N-1} \lambda_{q,j}\, (\Sigma_j\, \zeta_z^j)_i\, (\delta\Sigma_j)_{i,q} \\
&= \sum_{z=k}^{p} \sum_{i=1}^{N-1} \lambda_{i,j}\, (\Sigma_j\, \zeta_z^j)_i\, \delta\bar{\sigma}_{i,j} \\
&= \sum_{i=1}^{N-1} \sum_{z=k}^{p} (\Lambda_j\, \Sigma_j\, \zeta_z^j)_i\, \delta\bar{\sigma}_{i,j}
\end{aligned}$$

Note that Σ_j and $\delta\Sigma_j$ are diagonal matrices, i.e. if $(\Sigma_j)_{i,q}$ and $(\delta\Sigma_j)_{i,q}$ denote the element in the ith row and the qth column of Σ_j and $\delta\Sigma_j$, respectively, then $(\Sigma_j)_{i,q} = (\delta\Sigma_j)_{i,q} = 0$ for $i \neq q$. Hence, in the fourth row in the previous transformations, all terms in the sum with $i \neq q$ are zero.

Finally, it follows from the previous equation that

$$\tilde{\sigma}_j = \Lambda_j \, \Sigma_j \sum_{z=k}^{p} \zeta_z^j \,, \quad \text{for} \quad j = j_{k-1}, \ldots, j_k - 1 \,, \quad k = 1, \ldots, p \,,$$

which concludes the proof. ■

The computational effort necessary for the determination of the product $J(\bar{\sigma})^T \delta c = \tilde{\sigma}$ consists of setting up the matrices $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$ and the vectors λ_j for $j = 0, \ldots, M-1$, the determination of the parameters ζ_k^j, $k = 1, \ldots, p$, $j = j_k - 1, \ldots, 0$, and the subsequent computation of $\tilde{\sigma}_j$, $j = 0, \ldots, M-1$. As in the determination of the computational effort for the calculation of the product of $J(\bar{\sigma})\delta\bar{\sigma}$, the computational effort for setting up the matrices $U_j(\bar{\sigma})$ and $V_j(\bar{\sigma})$ for $j = 0, \ldots, M-1$ is given by

$$\begin{aligned}
&\text{computation of } \tilde{D}_j(\bar{\sigma}) : \quad && 6\,(N-2) + N = 7\,N - 12 \quad \text{multiplications}\,,\\
&\text{computation of } U_j(\bar{\sigma}) : && 3\,N - 5 \quad \text{multiplications}\,,\\
&\text{computation of } V_j(\bar{\sigma}) : && 3\,N - 5 \quad \text{multiplications}\,.
\end{aligned}$$

Moreover, the number of multiplications necessary to compute λ_j is $6N - 5$ for each $j = 0, \ldots, M-1$. The computation of ζ_k^j is done by the backward recursion (5.5.19) and the solution of the linear tridiagonal system of equations (5.5.20). For each k the initial setting requires $(N-1)\, m_{j_k}$ multiplications. In the subsequent iteration of this backward recursion process, $(3N-5)\, j_k$ multiplications are necessary for the j_k matrix–vector products $\mu_k^j = V_j(\bar{\sigma})^T \zeta_k^j$ and, additionally, $(5\,N - 9) j_k$ multiplications have to be carried out for the solution of the linear tridiagonal systems $U_j(\bar{\sigma})^T \zeta_k^j = \mu_k^{j+1}$. Finally, the computation of each $\tilde{\sigma}_j$, $j = 0, \ldots, M-1$, makes $2\,(N-1)$ multiplications necessary. Thus, the complete computational effort for one evaluation of $J(\bar{\sigma})^T \delta c$ amounts to

$$\begin{aligned}
&M\ (13\,N - 22 + 6\,N - 5 + 2N - 2) + m\,(N-1) + \sum_{k=1}^{p} j_k \cdot (5\,N - 9 + 3N - 5)\\
&= \ M\ (21\,N - 29) + m\,(N-1) + \sum_{k=1}^{p} j_k \cdot (8\,N - 14) \qquad (5.5.27)\\
&\le \ M\ (21\,N - 29 + p\ (8N - 14)) + m\,(N-1) \quad \text{multiplications}\,,
\end{aligned}$$

so that is of order $\mathcal{O}(pMN)$.

5.6 Numerical Results

In this last section, we present some computational experience with Algorithm 2.2.3 applied to problem (5.4.62) of constructing the instantaneous volatility function $\bar{\sigma}$. The model European call prices occurring in the residual function $R(\bar{\sigma})$ (5.4.61) are computed using the finite difference scheme (5.4.28) – (5.4.31) with $\theta = \frac{1}{2}$, i.e. the Crank–Nicolson method. For the finite difference scheme (5.4.28) – (5.4.31), we employ a uniformly spaced mesh with $N \times M$ grid points in the region $[\nu_1 \, S \, , \nu_2 \, S] \times [0 \, , \bar{T}]$ where $\bar{T}$ is the maximum maturity of the market option data. The evaluation of the Jacobian $J(\bar{\sigma})$ of the residual function $R(\bar{\sigma})$ times a vector and the transpose of this Jacobian times a vector are carried out according to Theorem 5.5.2 and Theorem 5.5.3, respectively. Moreover, the linear systems appearing in the Modified Preconditioned Conjugate Gradient Method 2.2.2 are solved by applying the singular value decomposition to the scaling matrix $\bar{D}^{1/2}$. Note that the singular value decomposition of $\bar{D}$ must only computed once since the scaling matrix $\bar{D}$ is constant for the complete optimization process.

We divide this section into two parts. In the first part, we demonstrate the behavior of the algorithm by constructing the instantaneous volatility function in several synthetic European call option examples. In these examples we pretend to know the instantaneous volatility function. Then, in the second part, we apply the algorithm to real world problems by examining European DAX index call options.

5.6.1 Synthetic European Call Option Examples

For the demonstration of the applicability of Algorithm 2.2.3 onto the determination of the volatility parameter $\bar{\sigma}$, we start our examination with a synthetic example. We assume that the asset underlying the option follows the stochastic differential equation

$$dS = \mu(S,t) \, S \, dt + \sigma^*(S,t) \, S \, dW_t \, , \quad t \in [0, \bar{T}]$$

such that the price function with respect to the strike price K and the maturity T of the European call option follows the deterministic parabolic differential equation

$$C_T - \tfrac{1}{2}\sigma^*(K,T)^2 K^2 C_{KK} + (r(T) - d(T))KC_K + d(T)C = 0 \tag{5.6.1}$$

subject to the initial condition

$$C(\sigma(\cdot,\cdot); K, 0) = \max(S - K, 0) \, .$$

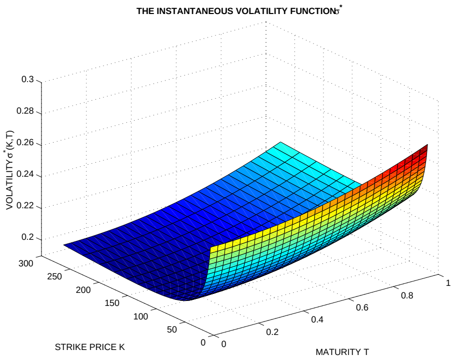


FIGURE 5.4: The instantaneous volatility function $\bar{\sigma}^*$

Moreover, we suppose that the instantaneous volatility function $\sigma^*(K,T)$ in (5.6.1) is given by

$$\sigma^*(K,T) = 0.2 + 0.005\,(\ln(\tfrac{K}{S}))^2 + 0.03\,T^2\,, \quad \text{for } (K\,,T) \in I\!R^+ \times [0\,,\bar{T}]\,. \tag{5.6.2}$$

Furthermore, we set

$$\begin{aligned} r(T) &= 0.05\,, \quad \text{for} \quad T \in [0\,,\bar{T}]\,,\\ d(T) &= 0.02\,, \quad \text{for} \quad T \in [0\,,\bar{T}] \end{aligned}$$

and assume that the current stock value is $S = 100$. Then, using these parameters, the corresponding European call option price function $C^*(K\,,T)$ is computed using the finite difference scheme (5.4.28) – (5.4.31) applied to the transformed version (5.2.12) of the parabolic differential equation (5.6.1) respective the parabolic initial–boundary–value problem (5.4.6) with $[x_{\min}\,,x_{\max}] = [\nu_1\,S\,,\nu_2\,S]$ where $\nu_1 = 0.05$ and $\nu_2 = 3$. The number of grid points with respect to the space variable x and the time variable T were chosen to be $N = M = 30$.

Since we have carried out this computation with the "exact" instantaneous volatility function $\sigma^*(K,T)$ given in (5.6.2) in the following examples, we interpret a certain number of these European call option values as market European call option data, i.e. $\hat{C}_{K_l,T_l} = C^*(S\,\exp(x_l)\,,T_{j_l})$, $l = 1\,,\ldots,m$, and

$$c_{observ} = (\hat{C}_{K_1,T_1}\,,\ldots,\hat{C}_{K_m,T_m})^T \in I\!R^m$$

Furthermore, the vectors with the corresponding strike prices and maturities, respectively, are designated as

$$\begin{aligned} K_{observ} &= (K_1\,,\ldots,K_m)^T \in I\!R^m\,,\\ T_{observ} &= (T_1\,,\ldots,T_m)^T \in I\!R^m\,. \end{aligned}$$

We will see in the following subsection that usually only for a limited number of strike prices and only few maturity dates options are available in the market so that in general only a small number of observations exist in comparison to the large number of unknowns that have to be estimated. Ignoring this aspect for a moment, we start our examination by assuming that at each grid knot in the chosen grid of the finite difference scheme there exists market European call option values, i.e.

$$c_{observ} = (C^*_{1,1}\,,\ldots,C^*_{N-1,1}\,,\ldots,C^*_{1,M}\,,\ldots,C^*_{N-1,M})^T \in I\!R^{(N-1)M}$$

with

$$C^*_{i,j} = C^*(S\,\exp(x_i)\,,T_j)\,.$$

We started the optimization routine with an initial guess of the volatility function of $\bar{\sigma}^0 = 0.24 = const.$ The computation of the European call option values appearing in the residual function $R(\bar{\sigma})$ is carried out by using the same discretization parameters N and M in the finite difference scheme as in the computation of the European call values $\hat{C}_{K_l,T_l} = C^*(S\,\exp(x_l)\,,T_{j_l})$, $l = 1\,,\ldots,m$. The algorithm stopped after 12 Gauss–Newton iterations with a computed optimal objective function value $\frac{1}{2}\|R(\bar{\sigma}^{opt})\|$ of 10^{-14} with $\bar{\sigma}^{opt}$ is the approximated volatility function computed by Algorithm 2.2.3. The volatility function $\bar{\sigma}^{opt}$ and its deviation from

$$\bar{\sigma}^* = (\hat{\sigma}^*_{1,0}\,,\ldots,\hat{\sigma}^*_{N-1,0}\,,\ldots,\hat{\sigma}^*_{1,M-1}\,,\ldots,\hat{\sigma}^*_{N-1,M-1})^T$$

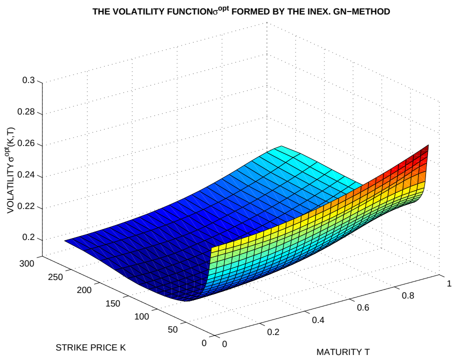


FIGURE 5.5: Volatility function $\bar{\sigma}^{opt}$ determined by the Inexact Gauss–Newton Method 2.2.3

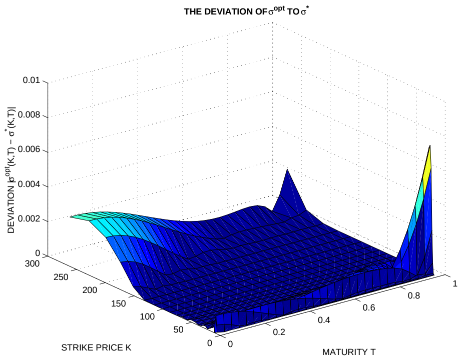


FIGURE 5.6: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^{*}$

with

$$\hat{\sigma}^*_{i,j} = \sigma^*(S\,\exp(x_i)\,,T_j)$$

are shown in Figure 5.5 and 5.6, respectively. These two figures show that in this case the accuracy of the approximated solution is acceptable. For a more precise demonstration of the accuracy we have plotted the volatility function $\bar{\sigma}^{opt}$ against the "true" instantaneous volatility function $\bar{\sigma}^*$ for the maturities $T = 0\,, 0.5$ and $T = 1$ on the left hand side in Figure 5.9. If observation data are available at every grid point in the finite difference grid, the error of the approximated solution is very small. We observe that for $T = 0.5$ the solution and its approximation are almost identical and that they only slightly deviate for $T = 0$ in case of larger strike prices and for $T = 1$ for smaller strike prices.

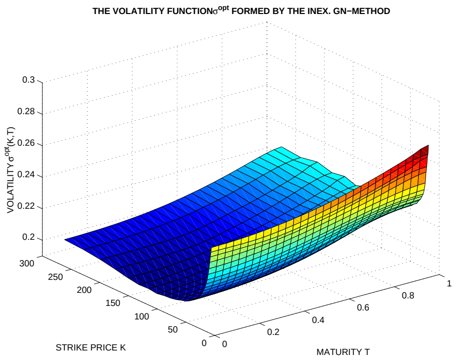


FIGURE 5.7: Volatility function $\bar{\sigma}^{opt}$ determined by the Inexact Gauss–Newton Method 2.2.3

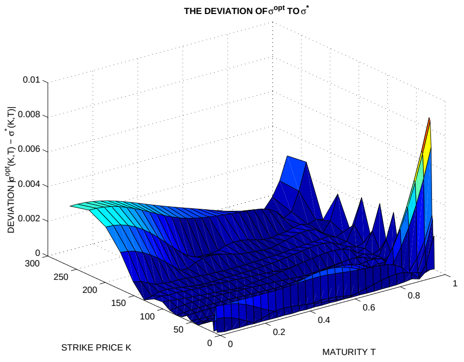


FIGURE 5.8: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^{*}$

Since the assumption of the existence of market European call data at every grid point of the finite difference grid is not very realistic, next, we successively reduce the number of observable option values.

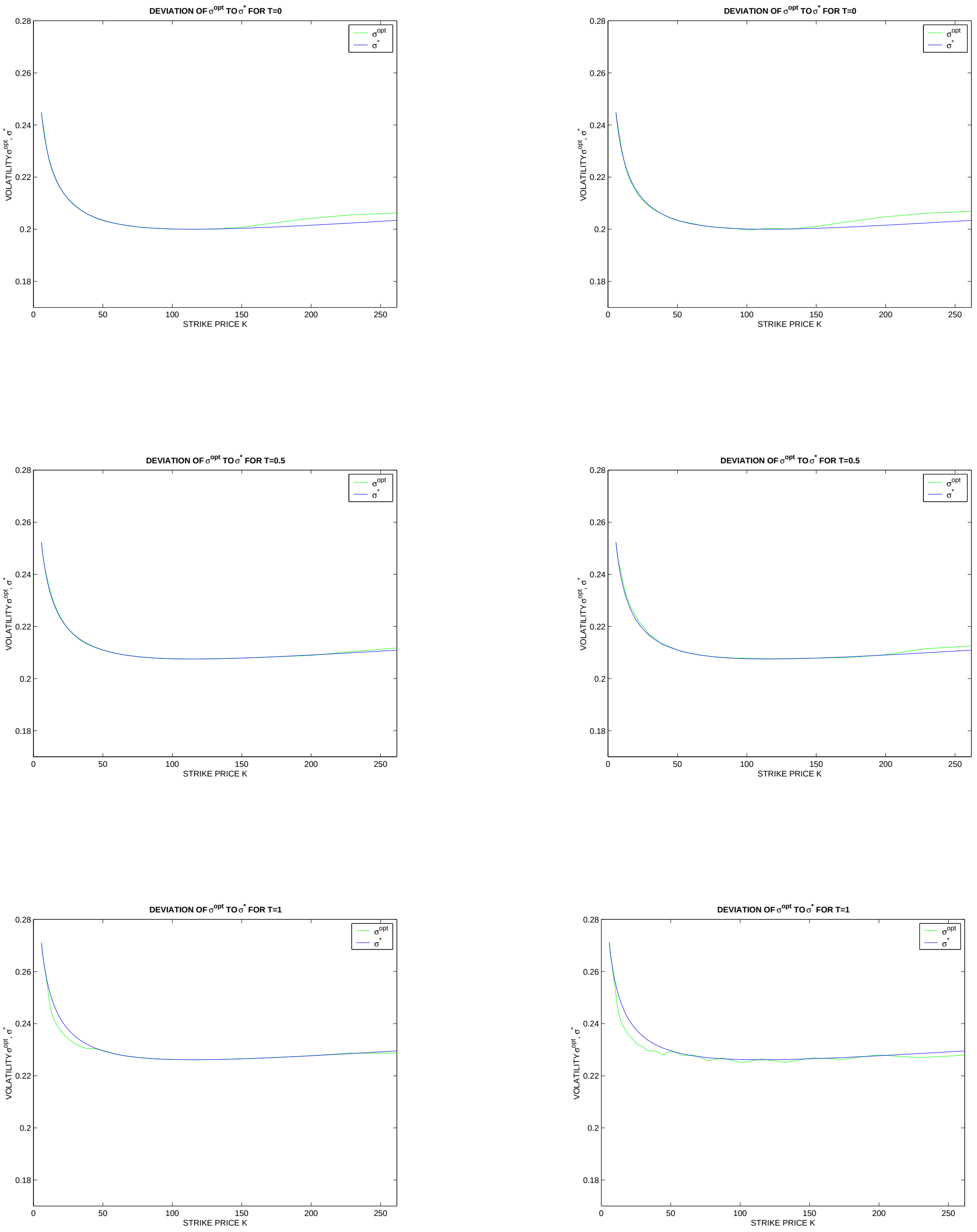


FIGURE 5.9: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^{*}$ for the maturities $T = 0\,, 0.5$ and $T = 1$; Left hand side: observations available at every grid point; Right hand side: observations available at every second grid point

Figures 5.7 and 5.8 show the constructed volatility function $\bar{\sigma}^{opt}$ and its deviation to $\bar{\sigma}^*$ when market European call option values are available only at every second grid point. In this case the Inexact Gauss–Newton Method 2.2.3 needs 13 outer iteration to stop with a computed objective function value $\frac{1}{2}\|R(\bar{\sigma}^{opt})\|$ of 10^{-14}. The deviation of the approximated solution $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$ is still of order 10^{-3}, however, the plots on the right hand side of Figure 5.9 show that they are slightly larger than in the previous example.

Finally, we examine the more realistic case by assuming that only a small number of European call options with the same underlying are traded. Figures 5.10 and 5.11 contain the results when European call options are available with six different maturities and each series having eight different strike prices, i.e. 48 observed European call options are considered as market data. The maturities are $T = 0.1\,, 0.17\,, 0.23\,, 0.5\,, 0.76$ and $T = 1$ and for the strike prices K_i, $i = 1\,\ldots, 8$, holds $K_i \in [0.5\,S\,, 1.3\,S]$ such that the vector with the observed European call values is given by

$$c_{observ} = (C_4^*\,, C_6^*\,, C_8^*\,, C_{16}^*\,, C_{24}^*\,, C_{31}^*)^T \in I\!R^{48} \tag{5.6.3}$$

with

$$C_j^* = (C^*(S\,\exp(x_{18})\,, T_j)\,, \ldots\,, C^*(S\,\exp(x_{25})\,, T_j))^T \in I\!R^8\,, \quad j = 4\,, 6\,, 8\,, 16\,, 24\,, 31\,.$$

Regarding the finite difference grid, this means that the observation data are located only in a small strip of the grid with respect to the state variable x, i.e. for small and large strike prices there are no observation data available.

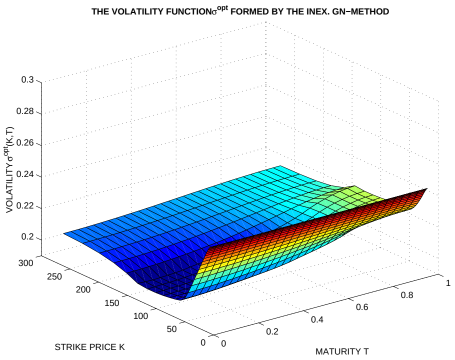


FIGURE 5.10: Volatility function $\bar{\sigma}^{opt}$ determined by the Inexact Gauss–Newton Method 2.2.3

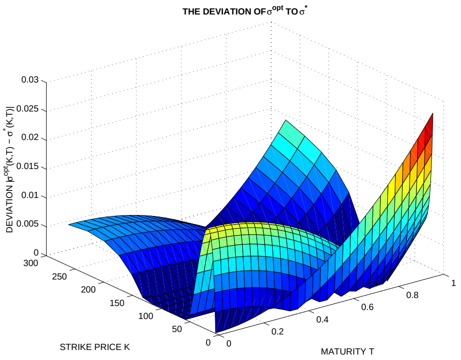


FIGURE 5.11: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$

For this example, the Inexact Gauss–Newton Method 2.2.3 requires 50 Gauss–Newton iterations to stop with the same size of the objective function than in the previous examples. Thus, the number of iterations has increased considerably in comparison with the previous two examples. Since the number of market European call values has been decreased the nonlinear least–squares problem (5.4.62) became more underdetermined, and thus, is more difficult to solve. Moreover, Figures 5.10 and 5.11 show that the quality of the approximation of $\bar{\sigma}^{opt}$ is worse than for the first two examples. For a more precise examination of the deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$, we have plotted $\bar{\sigma}^{opt}$ and $\bar{\sigma}^*$ for certain fixed maturities in Figure 5.12. These plots show that in the region where observation data are available the volatility function $\bar{\sigma}^*$ is approximated very accurately by $\bar{\sigma}^{opt}$. However, in the region where no observations are available, i.e. for small and large strike prices, the deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$ is much larger.

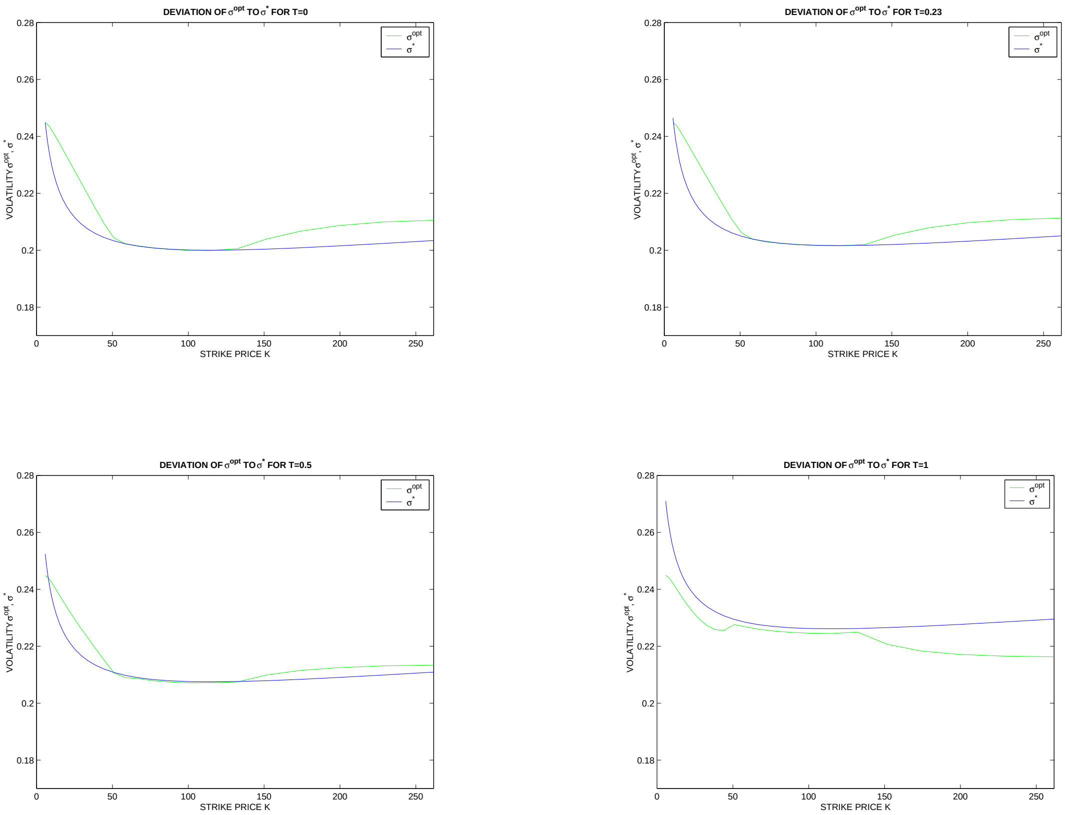


FIGURE 5.12: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$ for the maturities $T = 0\,,0.23\,,0.5$ and $T = 1$

For an explanation of this behavior, we examine the influence of different volatility values $\hat{\sigma}^*_{i,j-1}$ on the approximated European call option values $z_j(\bar{\sigma}^*)$. Figures 5.13 and 5.14 are showing the partial derivatives of the discrete approximation $z_j(\bar{\sigma}^*)$ of the transformed European call price function $c(x, T_j)$ with respect to the volatilities $\hat{\sigma}^*_{j-1}$ for a fixed maturity $j \in \{1\,,\dots,29\}$. In Figure 5.13, we have plotted the elements of the Jacobian $(\partial z_{i\,,20}(\bar{\sigma}^*)/\partial\hat{\sigma}^*_{k\,,19})_{1\le i\,,k\le 29}$. Since the size of these elements differ considerably, Figure 5.14 contains only the gradients of $z_{i\,,20}(\bar{\sigma}^*)$ with respect to $\hat{\sigma}^*_{19}$ for $i = 3,\dots,18$, i.e. in Figure 5.14 we have plotted the partial derivatives $(\partial z_{i\,,20}(\bar{\sigma}^*)/\partial\hat{\sigma}^*_{k\,,19})_{\substack{i=3,\dots,18\\k=1,\dots,29}}$. These two plots show that only a few volatility parameters $\hat{\sigma}^*_{k\,,19}$ seem to have an influence on the value of $z_{i,j}(\bar{\sigma}^*)$. For a more precise observation of this aspect, for $i = 1, 10, 20, 29$ we have drawn the gradients of $z_{i\,,20}(\bar{\sigma}^*)$ with respect to $\hat{\sigma}^*_{19}$ in Figure 5.15, i.e. they show the values of $(\partial z_{i,20}(\bar{\sigma}^*)/\partial\hat{\sigma}^*_{k,19})_{k=1,\dots,29}$ for $i = 1, 10, 20$ and 29, respectively. These last plots make clear that the main influence on the value of $z_{i,j}(\bar{\sigma}^*)$ is given by $\hat{\sigma}^*_{i,j-1}$ and that

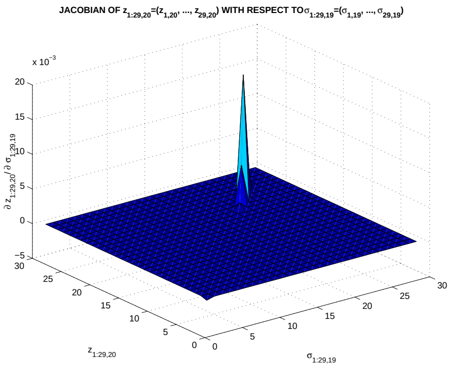


FIGURE 5.13: Partial derivatives of the European call values with maturity T, $z_{i,j}(\bar{\sigma}^*)$, $i = 1, \ldots, N$, with respect to the volatilities $\hat{\sigma}^*_{i,j-1}$ for $i = 1, \ldots, N$, respectively.

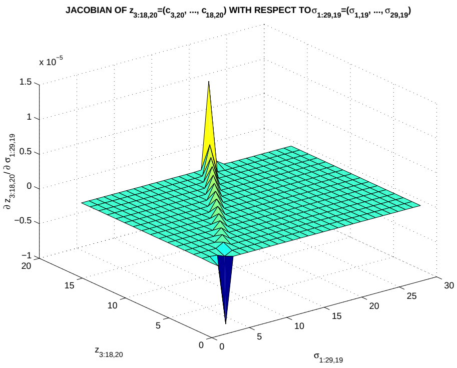


FIGURE 5.14: Partial derivatives of the European call values with maturity $\tilde{T}$, $z_{i,j}(\bar{\sigma}^*)$, $i = 1, \ldots, N$, with respect to the volatilities $\hat{\sigma}^*_{i,j-1}$ for $i = 1, \ldots, N$, respectively.

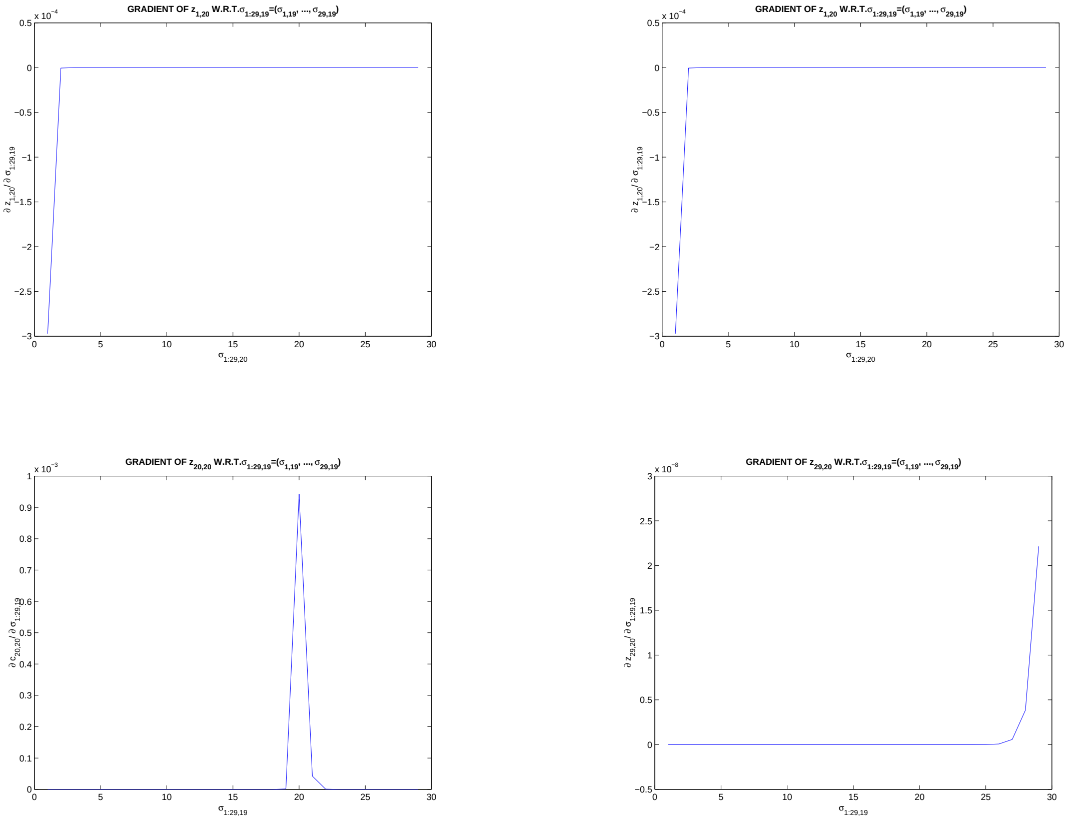
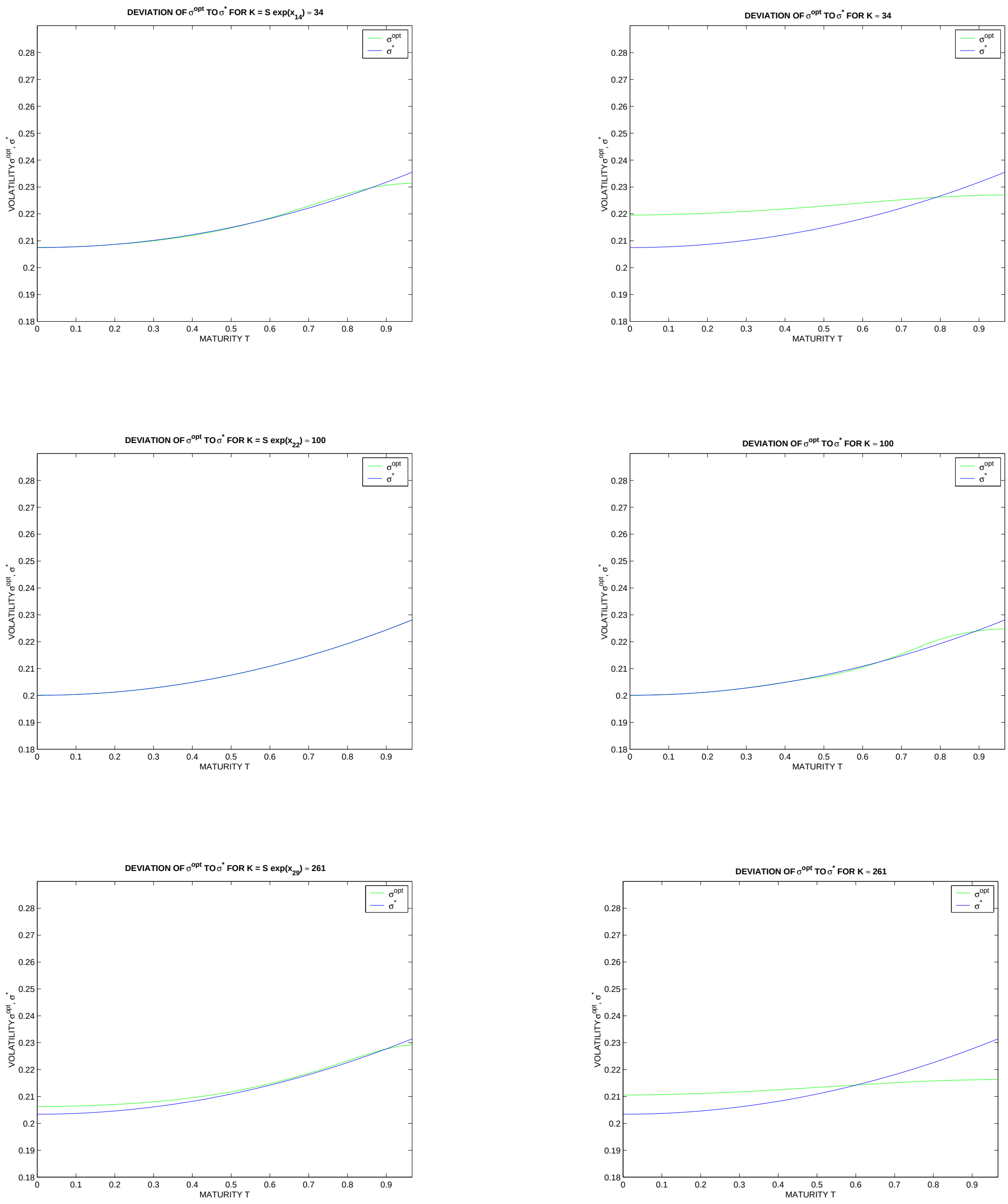


FIGURE 5.15: Gradient of $z(1,20)$, $z(10,20)$, $z(20,20)$ and $z(29,20)$ with respect to $\hat{\sigma}_{19}$, respectively

$z_{i,j}(\bar{\sigma}^*)$ is almost independent of $\hat{\sigma}^*_{k,j-1}$ for $k \neq i$. An explanation for this behavior might be the tree–like structure of finite difference methods. Thus, incorrect values of $\hat{\sigma}_{k,j-1}$ do not lead to a large error in the computation of $z_{i,j}(\bar{\sigma})$ for $k \neq i$, unfortunately, such that a determination of the volatility parameter $\bar{\sigma}$ can be carried out with an acceptable accuracy only in the region where market European call values are available.

So far, we have explored the deviation of the approximation $\bar{\sigma}^{opt}$ to its solution $\bar{\sigma}^*$ with respect to the strike price K. Next, we examine the behavior of the approximation $\bar{\sigma}^{opt}$ concerning the maturity T. Therefore, we have plotted $\bar{\sigma}^{opt}$ and $\bar{\sigma}^*$ for certain fixed strike prices K for the first and the third example on the left and on the right hand side, respectively, in Figure 5.16. The plots on the left hand side confirm the earlier statements of the high accuracy of the approximation $\bar{\sigma}^{opt}$ when observation data are available at every grid knot in the finite difference grid. On the right hand side of this figure, the plots are shown for the case if only a small number of observation data exist. Note that in this example, we have assumed that market European call options with strike price $K = S\ \exp(x_{23})$ were available; however, they do not exist for strike prices $K = S\ \exp(x_{15})$ and $K = S\ \exp(x_{29})$, respectively. The second plot on the right hand side of Figure 5.16 shows that for the former strike price the approximated volatility function σ^{opt} formed by the Inexact Gauss–Newton Method 2.2.3 almost coincides with its solution for all considered maturities; independent if market option values with these maturities exist or not. For the latter strike prices the approximation $\bar{\sigma}^{opt}$ shows a linear behavior in contrast to the quadratic behavior of its solution $\bar{\sigma}^*$, see the first and the last plot on the right hand side of Figure 5.16. Starting with an initial setting of $\bar{\sigma}^0 = 0.24$, the approximated solution $\bar{\sigma}^{opt}$ approaches $\bar{\sigma}^*$, however, is not able to identify the behavior of $\bar{\sigma}^*$ more precisely.



DEVIATION OF σ^{opt} TO σ^* FOR K = S exp(x_{14}) ≈ 34

DEVIATION OF σ^{opt} TO σ^* FOR K ≈ 34

DEVIATION OF σ^{opt} TO σ^* FOR K = S exp(x_{22}) ≈ 100

DEVIATION OF σ^{opt} TO σ^* FOR K ≈ 100

DEVIATION OF σ^{opt} TO σ^* FOR K = S exp(x_{29}) ≈ 261

DEVIATION OF σ^{opt} TO σ^* FOR K ≈ 261

FIGURE 5.16: Deviation of $\bar{\sigma}^{opt}$ to $\bar{\sigma}^*$ for the strike prices $K = S\,\exp(x_{15})$ (≈ 34), $S\,\exp(x_{23})$ (≈ 100) and $K = S\,\exp(x_{29})$ $(\approx$ 262); Left hand side: observations available at every grid point; Right hand side: 48 observations available at specified grid points, see (5.6.3)

5.6.2 Construction of the Instantaneous Volatility Function for the German Stock Index DAX

Finally, in this last subsection, we examine a more realistic example. We use values of market European DAX call options of February 2, 2002 to approximate the instantaneous volatility function $\hat{\sigma}(x, T)$ of the German stock index DAX.

For the determination of the discrete version $\bar{\sigma}$ of the instantaneous volatility function, we have 136 market European DAX call options available. These options belong to ten different series, i.e. they have ten different maturities which start with one month and reach up to five years. Each series consists of a different number of options which is higher for series with shorter maturities and smaller for series with longer maturities. The strike prices of the options of each of these series are scattered on the interval $[3,500; 8,000]$. The corresponding market value of the DAX index is $S = 5,003.24$.

For the numerical solution of the parabolic initial–boundary–value problem (5.4.6), we choose a finite difference mesh approximating the region $[0.1\,S\,,2\,S] \times [0\,,5]$ that consists of $N = 50$ discretization points for the spatial domain $[0.1\,S\,,2\,S]$ and $M = 56$ discretization points for the time variable T.

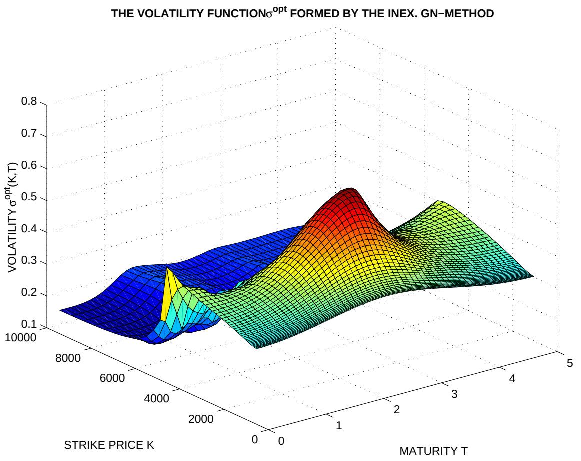


FIGURE 5.17: The approximated instantaneous volatility function $\bar{\sigma}$ of the German stock index DAX using all 136 data

We started the optimization process for the solution of the nonlinear least–squares problem (5.4.62) with an initial guess for the instantaneous volatility function of

$$\bar{\sigma}^0 = 0.28 = const\,.$$

Using all the given market European DAX call options, the Inexact Gauss–Newton Algorithm 2.2.3 stopped after 46 iterations with an approximated instantaneous volatility $\bar{\sigma}^{opt}$ for which the computed objective function value $\frac{1}{2}\,\|R(\bar{\sigma}^{opt}\|^2$ is of order 10^{-6}. The maximum absolute price error amounted to $2.0 \cdot 10^{-3}$ which occurred for prices belonging to options with long maturities. Thus, the given method excellently reproduces the market European DAX call values across the complete set of given data. The corresponding gradient $\|J(\bar{\sigma}^{opt})^T R(\bar{\sigma}^{opt})\|$ of the residual function R is of magnitude 10^{-5} such that we expect $\bar{\sigma}^{opt}$ to be a local minimizer. At the beginning of the optimization process, the Modified Preconditioned Conjugate Gradient Algorithm 2.2.2 needed only 2 iterations for the solution of the restricted Gauss–Newton subproblems (5.4.70), however, this number has slightly increased during the optimization process. The maximal number of conjugate gradient iterations required for the solution of these subproblems was 10 which is still very reasonable.

The approximated instantaneous volatility function $\bar{\sigma}^{opt}$ constructed by the given model and using all the 136 market European call option data is displayed in Figure 5.17. This picture shows that $\bar{\sigma}^{opt}$ exhibits a smooth behavior without any spikes. For sufficiently small and large strike prices K, the numerically determined values of $\bar{\sigma}^{opt}$ remain essentially unchanged over time. Regarding the spatial domain, this is the region where no market European DAX call options are located. We have already examined in the previous subsection that volatility values in these peripheral regions have little or no influence on an option value evaluated at a point that is located more in the interior of the domain $[0.1\,S\,,2\,S] \times [0\,,5]$.

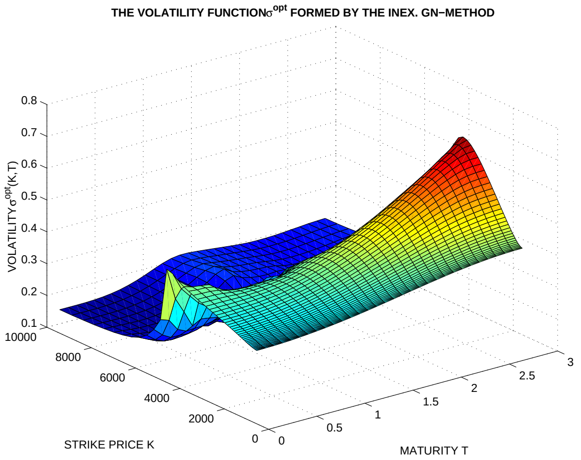


FIGURE 5.18: The approximated instantaneous volatility function $\bar{\sigma}$ of the German stock index DAX using all 129 data

Furthermore, we have determined the approximated instantaneous volatility function $\bar{\sigma}$ using all of the above explained market European DAX call options with a maturity less than three years, i.e. in this second computation we did not consider European DAX call options having a maturity of four and five years. Now, the data set consists of 129 market European DAX call options. For the numerical solution of the parabolic initial–boundary–value problem (5.4.6), we have used the same finite difference grid than in the previous example and have cut if off at maturity $T = 3$ years, i.e. the number of discretization points for the spatial domain and the time domain are given by $N = 50$ and $M = 33$, respectively.

In this case, the Inexact Gauss–Newton Method 2.2.3 applied to the nonlinear least–squares problem (5.4.62) required 40 iterations to yield a objective function value $\frac{1}{2}\|R(\bar{\sigma}^{opt})\|^2$ of order 10^{-6} and a value of the norm $\|J(\bar{\sigma}^{opt})^T R(\bar{\sigma}^{opt})\|$ of the gradient of order 10^{-5}, i.e. the same magnitudes than in the first computation using all of the 136 market European call option data. The maximum absolute error between the model and the market European call values is again $2.0 \cdot 10^{-3}$ which, this time, took place for options with maturities of three years. Thus, it seems that for the proposed method it is more difficult to identify the values of options with the longest maturity in the data set. Moreover, we again made the observation that at the beginning of the optimization process the number of iterations of the Modified Preconditioned Conjugate Gradient Method 2.2.2 required less iterations than at its end. However, compared to the dimension of the range space of the residual function of 129 the maximum number of 8 conjugate gradient iterations for the solution of the Gauss–Newton subproblems (5.4.70) is again reasonable.

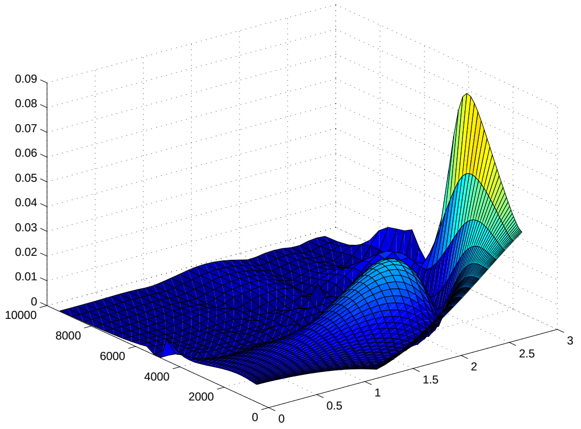


FIGURE 5.19: Deviation of the instantaneous volatility function $\bar{\sigma}$ given in Figures 5.17 and 5.18 for maturities up to 3 years

Finally, we compare the instantaneous volatility functions $\bar{\sigma}$ constructed with the complete data set consisting of 136 option values and with the reduced data set containing 129 option data, respectively. In Figure 5.19, we display the deviation of these two approximated instantaneous volatility functions given in Figures 5.17 and 5.18 for all values of $\hat{\sigma}_j$ with maturities $T_j \leq 3$ years. This picture makes clear that the two approximated instantaneous volatility functions $\bar{\sigma}$ almost coincide for large strike prices independent of the maturity. For short maturities, the deviation increases only slightly with decreasing strike price K. However, for larger maturities, the deviation of these two instantaneous volatility functions $\bar{\sigma}$ gets larger for small strike prices. Thus, the largest deviations occur for volatility values with strike prices smaller than the current value of the underlying and a maturity of three years. This observation corresponds to the above given speculation that the given model has more difficulties identifying option values with longer maturities.

Chapter 6

Conclusions

In this work, we were concerned with the identification of parameters for large scale underdetermined systems arising in finance. We have shown that these problems can be described by underdetermined nonlinear least–squares problems.

For this class of problems, it is likely that the residual function R at the solution is fairly small. It is well known that in this case the Gauss–Newton method exhibits a good asymptotic convergence behavior so that we have decided to choose this method for the solution of the nonlinear least–squares problems. To obtain global convergence of this algorithm, we applied a trust–region strategy to the Gauss–Newton method so that in each iteration of the Gauss–Newton algorithm a restricted linear least–squares problem has to be solved. Compared to line search methods, the trust–region strategy has the advantage that this method additionally enforces a regularization of the subproblems. This is especially in case of underdetermined residual functions useful since the solution of the linearized least–squares problems cannot be expected to be unique. Hestenes and Stiefel [56] presented, already in their original paper on conjugate gradients, a special version of the conjugate gradient method for linear least–squares problems. We applied this version of the conjugate gradient method to the solution of the linear Gauss–Newton subproblems. Furthermore, we included two additional stopping criteria presented by Toint [105] and Steihaug [98] into the preconditioned conjugate gradient method to take into account the restriction introduced by the trust–region strategy and the non–positive curvature of the Jacobian, since we cannot expect them to have a full column rank. In each iteration of the preconditioned conjugate gradient method, an evaluation of the matrix–vector products of the Jacobian of the residual function as well as its transpose times a vector has to be carried out. We have shown that for both applications in this work, we were able to make subroutines available that efficiently compute these matrix–vector products without saving the Jacobian explicitly. Thus, for the solution of the underdetermined nonlinear least–squares problems, we used a fully iterative inexact Gauss–Newton method.

Although the local convergence behavior of the Gauss–Newton method is already examined extensively, so far it did not exist any q–convergence results for underdetermined residual functions. In this work, we have shown that in general the q–linear convergence rate of the inexact Gauss–Newton sequence is obtained on the orthogonal complement of the null space of the Jacobian, i.e. on a linear subspace. In the case of zero residual problems, i.e. the residual function is zero at the solution, this convergence rate is even q–quadratically. Furthermore, for a special sequence of the Gauss–Newton method that can however only be chosen theoretically, we obtained the q–linear convergence on $I\!R^m$ for overdetermined as well as underdetermined residual functions. Moreover, we have seen that in case of overdetermined nonlinear least–squares problems, the presented results specialize to known results [50]. Note, that all the convergence results derived in this work are invariant under unitary transformations.

Subsequently, we have applied the inexact Gauss–Newton method to the solution of nonlinear least–squares problems arising from the estimation of parameters in neural networks and in the valuation of European call options.

In a neural network, the unknown parameters consist in the weights matrices between the different layers and the bias of the different neurons in the neural network. We have shown that the determination of these unknown weights and biases in the neural network, i.e. the training of the neural network, is described by a nonlinear least–squares problem. When neural networks are applied to problems in finance, usually the number of training patterns available to train the network is much smaller than would be needed to specify the associated dynamical system such that the number of unknown weights and bias is often much larger than the number of training pattern. Thus, for neural networks applied in forecasting, the nonlinear least–squares problem describing the training process of the neural network is underdetermined. An application of the proposed inexact Gauss–Newton method has led to a significant reduction of the computation time compared to the traditional method for the training process, the backpropagation algorithm. Moreover, the low number of required conjugate gradient iterations in the optimization process suggests a careful selection of the conjugate gradient iterates. Thus, the efficiency of the inexact Gauss–Newton method enables the treatment also of very large size networks with about 500,000 unknown weights and biases.

Furthermore, we have examined the ability of neural networks of forecasting the German stock index DAX. For an efficient forecast of a stock or index price, it is necessary to examine the linear and nonlinear dependencies of a large number of variables as well as their interactions. The advantage of the application of neural networks in forecasting is that the underlying function form which generates the data does not need to be known in advance; the neural network rather tries to find this function during the learning process. However, we have seen that complete freedom in the choice of the model is also not given if neural networks are used. A crucial part in the design of the neural network is the choice of the data set as well as the choice of the number of layers and the number of neurons in any of the layers in the neural network which proves to be an extensive trial–and–error process. To attack the problem of overfitting (the problem of fitting the noise in addition to the signal) and to achieve the ability of generalization of the neural network, we have introduced the Stopped–Training method. The proposed examples have shown that the rate of right trends and the performance of a trading strategy increases if the network is not only fed with DAX prices but also with other information which could have an influence on the development of the DAX price. Moreover, we showed that there exist networks whose trading strategy beats the benchmark quite clearly. However, we also noticed that the performance of this network was worse for other periods. Therefore, we introduced the backtesting which achieved good results too; however, they were also not generally obtained. These results elucidate the difficulties in finding an optimal neural network for the given task. Finally, we determined the sensitivities of the optimal output of the neural network with regard to small variations of the various components of the input vector such that we could at least derive some information about the influence of each of the input data on the output of the neural network.

In the last chapter, we were concerned with the valuation of European call options. Assuming that the underlying asset of the European call option follows a continuous one–factor diffusion model, we presented a method of approximating the instantaneous volatility function $\sigma(K,T)$ using a finite set of market European call prices. The model European call option prices were computed numerically by applying a finite difference method to the parabolic differential equation describing the price process of the European call option. The examination of the convergence of the finite difference scheme has shown that difficulties in the examination of the consistency of the finite difference scheme occur when the initial condition is not sufficiently smooth. Moreover, we have shown that the order of convergence of the solution of the finite difference scheme to the solution of the corresponding parabolic differential equation depends on the order of accuracy of a sufficiently smooth approximation of the nonsmooth initial condition of the initial–boundary–value problem describing the price process of the European call options.

The unknown instantaneous volatility values at the grid points in the finite difference mesh are determined by solving an underdetermined nonlinear least–squares problem. We have solved these underdetermined nonlinear least–squares problems using the proposed inexact Gauss–Newton method. To take account for the smoothness of the volatility parameter, the restrictions in the linear least squares problems are scaled with a modified discrete differentiation operator. For an illustration of the capability of the proposed method, we considered two European call option examples. In the first example, we considered synthetic European call options for which the underlying follows a known 1–factor diffusion model. The market European call data were simulated by evaluating a set of European call options using a finite difference scheme applied to the parabolic differential equation containing the “exact” instantaneous volatility function. A comparison of the reconstructed instantaneous volatility function to the “exact” instantaneous volatility function indicates a fairly accurate construction of the volatility function in the region within which the market European call options are located. In the second example, European DAX index call options with market data of February 2002 are considered. The resulting constructed instantaneous volatility function exhibits a smooth behavior.

Our first computations for the determination of the instantaneous volatility function were carried out in Matlab. They show a promising behavior of the proposed method. However, to take advantage of the complete efficiency of the proposed inexact Gauss–Newton method, it is necessary to code the program in a program language like C or C++.

Appendix A

The Parabolic Initial–Boundary–Value Problem

A.1 Existence and Uniqueness of the Solution

Friedman showed in [37] the existence of the initial boundary value problem $L\,c = c_T$ with initial condition (5.2.13) and boundary condition (5.4.5), using the method of continuity, i.e. by examining the family of parabolic operators

$$L_\lambda\, c = \lambda\,(L\,c - c_T) \;+\; (1-\lambda)\,(c_x x - c_T)\,, \quad 0 \le \lambda \le 1\,,$$

and showing that if the problem can be solved for all $\lambda < \sigma$ $(0 < \sigma \le 1)$ then it also can be solved for $\lambda \le \sigma$, and that if the problem can be solved for all $\lambda \le \sigma$ $(0 \le \sigma < 1)$ then it can also be solved for all $\lambda \le \sigma + \epsilon$ for some $\epsilon > 0$ and since, finally, the problem can be shown to be solvable for $c_{xx} - c_T = 0$, the same is true for $L\,c - c_T = 0$. Then, the uniqueness of the solution to the initial boundary value problems follows from the weak maximum principle, see [37, Chap. 2, Theorem 7].

Moreover, let $c^1(x,T)$ and $c^2(x,T)$ satisfy the initial–boundary value problem $L\,c = c_T$ with boundary conditions (5.4.5) and initial conditions $c_1(x,0) = \varphi_1(x)$ and $c_2(x,0) = \varphi_2(x)$, respectively, with $\varphi_1(x)$ and $\varphi_2(x)$ are continuous functions. Then, the difference $c^3(x,T) = c^1(x,T) - c^2(x,T)$ is the solution to the parabolic differential equation $L\,c = c_T$ with homogenous boundary conditions and initial condition $c^3(x,0) = \varphi^1(x) - \varphi^2(x)$ and is continuous in $[x_{\min}\,, x_{\max}] \times [0\,,\bar{T}]$. Supposing $\hat{\sigma}^2 : [x_{\min}\,, x_{\max}] \times [0\,,\bar{T}] \to I\!R^+$ and $r\,, d : [0\,,\bar{T}] \to I\!R^+$ are continuous functions with $\hat{\sigma}^2(x,T)$ satisfying

$$\lambda_0 \le \hat{\sigma}^2(x,T) \le \lambda_1 \quad \text{for all} \quad (x,T) \in [x_{\min}\,, x_{\max}] \times [0, \bar{T}]\,,$$

then the weak maximum principle holds stating that the maximum of $c^3(x,T)$ in $[x_{\min}\,, x_{\max}] \times [0\,,\bar{T}]$ is attained on the boundary

$$[x_{\min}\,, x_{\max}] \times \{0\} \cup \{x_{\min}\} \times (0\,,\bar{T}] \cup \{x_{\max}\} \times (0\,,\bar{T}]\,,$$

see for instance Friedman [37, Chap. 2, Theorem 6]. Thus,

$$\begin{aligned}
|c^1(x,T) - c^2(x,T)| &= |c^3(x,T)| \\
&\le \max_{[x_{\min}\,,x_{\max}]\times[0\,,\bar{T}]} |c^3(x,T)| \\
&\le \max_{\substack{[x_{\min}\,,x_{\max}]\times\{0\} \\ \{x_{\min}\,,x_{\max}\}\times(0\,,\bar{T}]}} |c^3(x,T)| \\
&= \max_{[x_{\min}\,,x_{\max}]} |c^3(x,0)| \\
&= \max_{[x_{\min}\,,x_{\max}]} |\varphi^1(x) - \varphi^2(x)|\,.
\end{aligned}$$

Theorem A.1.1 *Let $\hat{\sigma}^2(x,T)$ be a continuous function in $[x_{\min}\,,x_{\max}]\times[0\,,\bar{T}]$, $r(T)$ and $d(T)$ continuous functions in $[0\,,\bar{T}]$ and let $\hat{\sigma}^2(x,T)$ satisfy*

$$\lambda_0 \le \hat{\sigma}^2(x,T) \le \lambda_1 \quad \textit{for all} \quad (x,T) \in [x_{\min}\,,x_{\max}]\times[0,\bar{T}]\,.$$

Then, there exists at most one solution to the parabolic initial–boundary value problem $L\,c = c_T$ with initial condition (5.2.13) *and boundary condition* (5.4.5)*. Moreover, the solution $c(x,T)$ depends continuously on the initial data, i.e. if $c_1(x,T)$ and $c_2(x,T)$ are solutions to the parabolic initial–boundary value problem $L\,c = c_T$ with boundary conditions* (5.4.5) *and initial conditions $c_1(x,0) = \varphi_1(x)$ and $c_2(x,0) = \varphi_2(x)$, respectively, with $\varphi_1(x)$ and $\varphi_2(x)$ are continuous functions, then*

$$|c_1(x,T) - c_2(x,T)| \le \max_{x\in[x_{\min}\,,x_{\max}]} |\varphi_1(x) - \varphi_2(x)|\,.$$

Hence, according to the definition of well–posedness by Hadarmard (see for instance Louis [70]), the initial–boundary value problem $L\,c = c_T$ with initial condition (5.2.13) and boundary condition (5.4.5) is a well–posed problem.

Moreover, we cite the existence result by Ladyženskaja, Solonnikov, and Ural'ceva [66, IV, Theorem 5.2] showing that the smoothness of the solution $c(x,T)$ to the parabolic differential equation (5.2.12) depends on the smoothness of the initial condition accompanying the differential equation, i.e. the order of differentiability of the solution $c(x,T)$ of the parabolic differential equation (5.2.12) increases with the order of differentiability of its initial condition.

For this, we first introduce the corresponding functional spaces which are used in this result. Let $H^l([x_{\min}\,,x_{\max}])$, l is a nonintegral positive number, be the Banach space whose elements are continuous functions $\varphi(x)$ in $(x_{\min}\,,x_{\max})$ having in $[x_{\min}\,,x_{\max}]$ continuous derivatives up to order $[l]$ inclusively and a finite value for the quantity ($\Omega = (x_{\min}\,,x_{\max})$)

$$|\varphi|_\Omega^{(l)} = \langle\varphi\rangle_\Omega^{(l)} + \sum_{j=0}^{[l]} \langle\varphi\rangle_\Omega^{(j)}$$

where

$$\langle\varphi\rangle_{\Omega}^{(0)} = |\varphi|_{\Omega}^{(l)} = \max_{\Omega} |\varphi| \,,$$

$$\langle\varphi\rangle_{\Omega}^{(j)} = \sum_{(j)} |D_x^j\varphi|_{\Omega}^{(0)} \,, \qquad \langle\varphi\rangle_{\Omega}^{(l)} = \sum_{([l])} \langle D_x^{[l]}\varphi\rangle_{\Omega}^{(l-[l])} \,,$$

$$\sup_{\substack{x,x'\in\Omega \\ |x-x'|\le\rho_0}} \frac{|\varphi(x)-\varphi(x')|}{|x-x'|^{\alpha}} \,, \quad 0<\alpha<1 \,.$$

and let $H^{l,l/2}([x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$ be the Banach space of functions $c(x,T)$ that are continuous in $[x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$, together with all derivatives of the form $D_T^r\,D_x^s$ for $2r+s<l$, and have a finite norm ($Q_{\bar{T}} = (x_{\min}\,,x_{\max}) \times (0\,,\bar{T})$)

$$|c|_{Q_{\bar{T}}}^{(l)} = \langle c\rangle_{Q_{\bar{T}}}^{(l)} + \sum_{j=0}^{[l]} \langle c\rangle_{Q_{\bar{T}}}^{(j)}$$

where

$$\langle c\rangle_{Q_{\bar{T}}}^{(0)} = |c|_{Q_{\bar{T}}}^{(l)} = \max_{Q_{\bar{T}}} |c| \,,$$

$$\langle c\rangle_{Q_{\bar{T}}}^{(j)} = \sum_{(2r+s=j)} |D_T^r\,D_x^s c|_{Q_{\bar{T}}}^{(0)} \,,$$

$$\langle c\rangle_{Q_{\bar{T}}}^{(l)} = \langle c\rangle_{x,Q_{\bar{T}}}^{(l)} + \langle c\rangle_{T,Q_{\bar{T}}}^{(l/2)} \,,$$

$$\langle c\rangle_{x,Q_{\bar{T}}}^{(l)} = \sum_{(2r+s=[l])} \langle D_T^r\,D_x^s c\rangle_{x,Q_{\bar{T}}}^{(l-[l])} \,,$$

$$\langle c\rangle_{T,Q_{\bar{T}}}^{(l/2)} = \sum_{0<l-2r-s<2} \langle D_T^r\,D_x^s c\rangle_{T,Q_{\bar{T}}}^{((l-2r-s)/2)} \,,$$

$$\langle c\rangle_{x,\bar{Q}_{\bar{T}}}^{(\alpha)} = \sup_{\substack{(x,T),(x',T)\in Q_{\bar{T}} \\ |x-x'|\le\rho_0}} \frac{|c(x,T)-c(x',T)|}{|x-x'|^{\alpha}} \,, \quad 0<\alpha<1 \,,$$

$$\langle c\rangle_{T,\bar{Q}_{\bar{T}}}^{(\alpha)} = \sup_{\substack{(x,T),(x,T')\in Q_{\bar{T}} \\ |T-T'|\le\rho_0}} \frac{|c(x,T)-c(x,T')|}{|T-T'|^{\alpha}} \,, \quad 0<\alpha<1 \,.$$

Then, the result on the unique solvability of the initial–boundary–value problem given by Ladyženskaja, Solonnikov, and Ural'ceva [66, IV, Theorem 5.2] is

Theorem A.1.2 *Suppose $l > 0$ is a nonintegral number and the coefficients of the operator L belong to the class $H^{l,l/2}([x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$. Then, for any function $\varphi(x) \in H^{l+2}([x_{\min}\,,x_{\max}])$ describing the initial condition the initial–boundary value problem has a unique solution from the class $H^{l+2,l/2+1}([x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$. It satisfies the inequality*

$$|c|_{Q_{\bar{T}}}^{(l+2)} \le k_1 \left(|\varphi|_{\Omega}^{(l+2)} + |c(x_{\min},T)|_{x_{\min}\times[0\,,\bar{T}]}^{(l+2)}\right)$$

Appendix B

Consistency of the Finite Difference Scheme

B.1 Proof of Theorem 5.4.10

In this section, we present the proof of Theorem 5.4.10 in Section 5.4 showing that the finite difference scheme (5.4.28) – (5.4.31) applied to the slightly perturbed initial–boundary–value problem is consistent of order $(2,1)$.

Theorem B.1.1 *Let the initial condition $\varphi(x)$ of* (5.4.43) *satisfy $\varphi \in H^{4+\epsilon}([x_{\min}\,, x_{\max}])$, with $\epsilon > 0$ arbitrary small. Then, the finite difference scheme* (5.4.28) – (5.4.31) *is consistent to the perturbed initial–boundary–value problem* (5.4.43) *of order $(2,1)$ with respect to $\| \cdot \|_{2,\Delta x}$ for $0 \leq \theta \leq 1$. Moreover, if $\varphi \in H^{6+\epsilon}([x_{\min}\,, x_{\max}])$, $\epsilon > 0$ arbitrary small, and the coefficients $b(T)$, $\hat{\sigma}(x,T)$ and $d(T)$ in the parabolic differential equation* (5.2.12) *are not dependent on T, i. e.*

$$b(T) = \bar{b}\,, \quad \hat{\sigma}(x,T) = \bar{\sigma}(x)\,, \quad d(T) = \bar{d}$$

then the finite difference scheme (5.4.28) – (5.4.31) *is consistent to the perturbed initial–boundary value problem* (5.4.43) *of order $(2,2)$ with respect to $\| \cdot \|_{2,\Delta x}$ for $\theta = \frac{1}{2}$, i.e. in this case the Crank–Nicolson method is consistent of order $(2,2)$ with respect to $\| \cdot \|_{2,\Delta x}$.*

Proof: Let $c(x,T)$ be the solution to the perturbed parabolic initial–boundary–value problem (5.4.43). According to the result by Ladyženskaja, Solonnikov, and Ural'ceva [66, Chap. IV, Theorem 5.2] the solution $c(x,T)$ belongs to $H^{4+\epsilon,(4+\epsilon)/2}([x_{\min}\,, x_{\max}] \times [0\,, \bar{T}])$ since $\varphi(x) \in H^{4+\epsilon}([x_{\min}\,, x_{\max}])$ (see Appendix A for the definition of the spaces $H^{4+\epsilon,(4+\epsilon)/2}([x_{\min}\,, x_{\max}] \times [0\,, \bar{T}])$ and $H^{4+\epsilon}([x_{\min}\,, x_{\max}])$ and the solvability result for the parabolic initial–boundary value problem presented by Ladyženskaja, Solonnikov, and Ural'ceva in [66]). Thus, all derivatives of $c(x,T)$ of the form $(\partial/\partial T)^r(\partial/\partial x)^s$, i.e. the rth derivative with respect to T and the sth derivative with respect to x, for $2r + s < 4 + \epsilon$ exist and are continuous.

In accordance to Definition 5.4.9, we have to verify that

$$\varepsilon_{i,j+1} \;=\; \tfrac{1}{\Delta T}\,(c_{i,j+1} - c_{i,j}) - \tilde{L}_{i,j}\,((1-\theta)\,c_{i,j} + \theta\, c_{i,j+1}) \to 0 \tag{B.1.1}$$

and that

$$\|\varepsilon_{j+1}\|_{2,\Delta x} \;=\; \mathcal{O}((\Delta x)^2) \;+\; \mathcal{O}(\Delta T)\,. \tag{B.1.2}$$

Using Taylor's theorem, we see that the forward difference quotient with respect to the time variable T satisfies

$$\tfrac{(c(x,T+\Delta T)-c(x,T))}{\Delta T} \;=\; c_T(x,T) + \mathcal{O}(\Delta T) \tag{B.1.3}$$

since the partial derivatives of $c(x,T)$ with respect to T on $[x_{\min}\,, x_{\max}] \times [0\,, \bar{T}]$ exist up to the order 2. Similarly, we get for the central difference quotient and the symmetric central difference quotient that

$$\tfrac{(c(x+\Delta x,T)-c(x-\Delta x,T))}{2\Delta x} \;=\; c_x(x,T) + \mathcal{O}((\Delta x)^2) \tag{B.1.4}$$

and

$$\tfrac{(c(x+\Delta x,T)-2c(x,T)+c(x-\Delta x,T))}{(\Delta x)^2} \;=\; c_{xx}(x,T) + \mathcal{O}((\Delta x)^2)\,. \tag{B.1.5}$$

because of the existence and continuity of $\frac{\partial^4}{\partial x^4} c(x,T)$ on $[x_{\min}\,, x_{\max}] \times [0\,, \bar{T}]$. Furthermore, the central difference quotient and the symmetric central difference quotient in the time layer $T + \Delta T$ can be written as

$$\begin{aligned} \tfrac{(c(x+\Delta x,T+\Delta T)-c(x-\Delta x,T+\Delta T))}{2\Delta x} &\;=\; c_x(x,T) + \mathcal{O}((\Delta T) + \mathcal{O}((\Delta x)^2) \\ \tfrac{(c(x+\Delta x,T+\Delta T)-2c(x,T+\Delta T)+c(x-\Delta x,T+\Delta T))}{(\Delta x)^2} &\;=\; c_{xx}(x,T) + \mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2) \end{aligned} \tag{B.1.6}$$

under the given smoothness characteristics of $c(x,T)$.

Now, plugging (B.1.3) – (B.1.6) into (B.1.1) and using the fact that $c(x,T)$ is a solution to the parabolic differential equation (5.2.12) yields for $i = N^- + 1\,, \ldots, N^+ - 1$ and

$j = 0\,,\dots,M-1$

$$\begin{aligned}
&\varepsilon_{i,j+1} \\
&\quad = \tfrac{1}{\Delta T}\,(c_{i,j+1} - c_{i,j}) - \tilde{L}_{i,j}\,((1-\theta)\,c_{i,j} + \theta\,c_{i,j+1}) \\
&\quad = \tfrac{1}{\Delta T}\,(c_{i,j+1} - c_{i,j}) - \Big[\tfrac{1}{2}\hat{\sigma}^2_{i,j}\,\Big((1-\theta)\,\tfrac{c_{i+1,j} - 2\,c_{i,j} + c_{i-1,j}}{(\Delta x)^2} + \theta\,\tfrac{c_{i+1,j+1} - 2\,c_{i,j+1} + c_{i-1,j+1}}{(\Delta x)^2}\Big) \\
&\qquad - \Big(b_j + \tfrac{1}{2}\hat{\sigma}^2_{i,j}\Big)\Big((1-\theta)\,\tfrac{c_{i+1,j} - c_{i-1,j}}{2\,\Delta x} + \theta\,\tfrac{c_{i+1,j} - c_{i-1,j}}{2\,\Delta x}\Big) \\
&\qquad - d_j\;\,((1-\theta)\,c_{i,j} + \theta\,c_{i,j+1})\big] \\
&\quad = c_T(i\,\Delta x\,,j\,\Delta T) - \tfrac{1}{2}\hat{\sigma}(i\,\Delta x\,,j\,\Delta T)^2\,((1-\theta)\,c_{xx}(i\,\Delta x\,,j\,\Delta T) + \;\theta\,c_{xx}(i\,\Delta x\,,j\,\Delta T)) \\
&\qquad + \;\big(b(j\,\Delta T) + \tfrac{1}{2}\hat{\sigma}(i\,\Delta x\,,j\,\Delta T)^2\big)\;((1-\theta)\,c_x(i\,\Delta x\,,j\,\Delta T) + \;\theta\,c_x(i\,\Delta x\,,j\,\Delta T)) \\
&\qquad + d(j\,\Delta T)\;\,((1-\theta)\,c(i\,\Delta x\,,j\,\Delta T) + \;\theta\,c(i\,\Delta x\,,j\,\Delta T)) \\
&\qquad + \mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2) \\
&\quad = \mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2)\,.
\end{aligned}$$

Since the perturbed initial–boundary–value problem (5.4.43) contains Dirichlet boundary conditions, we do not have to consider the boundary conditions separately since we have used the exact values for $c(N^-\,\Delta x\,,j\,\Delta T)$ and $c(N^+\,\Delta x\,,j\,\Delta T)$, $j = 0\,,\dots,M-1$, given by the functions describing the boundary conditions. Moreover, note that for $j = 0$ the condition

$$L\,c(x,0) \;=\; c_T(x,0)$$

is obtained by a sufficient smoothness of the coefficients appearing in the elliptic operator L and the smoothness of the solution $c(x,T)$ on $[x_{\min}\,,x_{\max}] \times [0\,,\bar{T}]$. For a detailed examination of the regularity we refer to Friedman [37, Chap. 10] or Wloka [108].

Furthermore, the discrete L_2–norm of the vector containing the truncation errors in the $j+1$st time layer fulfils

$$\begin{aligned}
\|\varepsilon_{j+1}\|^2_{2,\Delta x} &= \Delta x \sum_{i=N^-+1}^{N^+-1} \varepsilon^2_{i,j+1} \\
&= \Delta x \sum_{i=N^-+1}^{N^+-1} \big(\mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2)\big)^2 \\
&= (x_{\max} - x_{\min})\;\,\big(\mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2)\big)^2
\end{aligned}$$

such that

$$\|\varepsilon_{j+1}\|_{2,\Delta x} \;=\; \mathcal{O}(\Delta T) + \mathcal{O}((\Delta x)^2)\,.$$

Thus, if $c(x,T) \in H^{4+\epsilon,(4+\epsilon)/2}([x_{\min}\,,x_{\max}] \times [0\,,\bar{T}])$ the weighted finite difference scheme (5.4.28) is consistent of order $(2,1)$ for $0 \leq \theta \leq 1$.

In the following, we assume that $\varphi(x) \in H^{6+\epsilon}([x_{\min}\,, x_{\max}])$. Then, $c(x,T) \in H^{6+\epsilon,(6+\epsilon)/2}([x_{\min}\,, x_{\max}] \times [0\,, \bar{T}])$ and all derivatives of $c(x,T)$ of the form $(\partial/\partial T)^r(\partial/\partial x)^s$ for $2r + s < 6 + \epsilon$ exist and are continuous (see Appendix A). In this case, the Taylor expansion of the forward difference quotient with respect to the time T and the central difference quotient and the symmetric central difference quotient in the time layer $T + \Delta T$ with respect to the space variable can contain further terms such that (B.1.3) and (B.1.6) become

$$\frac{(c(x,T+\Delta T)-c(x,T))}{\Delta T} = c_T(x,T) + \frac{\Delta T}{2} c_{TT}(x,T) + \mathcal{O}((\Delta T)^2)\,, \tag{B.1.7}$$

and

$$\begin{aligned}
&\frac{(c(x+\Delta x,T+\Delta T)-c(x-\Delta x,T+\Delta T))}{2\Delta x}\\
&\quad = c_x(x,T+\Delta T) + \mathcal{O}((\Delta x)^2)\\
&\quad = c_x(x,T) + \Delta T\, c_{xT}(x,T) + \mathcal{O}((\Delta T)^2) + \mathcal{O}((\Delta x)^3)\,,\\
&\frac{(c(x+\Delta x,T+\Delta T)-2c(x,T+\Delta T)+c(x-\Delta x,T+\Delta T))}{(\Delta x)^2}\\
&\quad = c_{xx}(x,T+\Delta T) + \mathcal{O}((\Delta x)^2)\\
&\quad = c_{xx}(x,T) + \Delta T\, c_{xxT}(x,T) + \mathcal{O}((\Delta T)^2) + \mathcal{O}((\Delta x)^2)\,.
\end{aligned} \tag{B.1.8}$$

Using (B.1.4), (B.1.5), (B.1.7), and (B.1.8) then the local discretization error is given by

$$\begin{aligned}
\varepsilon_{i,j+1} = \ & \Delta T\ \Big(\tfrac{1}{2} c_{TT}(i\,\Delta x\,, j\,\Delta T) - \tfrac{1}{2}\hat{\sigma}(i\,\Delta x\,, j\,\Delta T)^2\,\theta\, c_{xxT}(i\,\Delta x\,, j\,\Delta T)\\
& + \big(b(j\,\Delta T) + \tfrac{1}{2}\hat{\sigma}(i\,\Delta x\,, j\,\Delta T)^2\big)\ \theta\, c_{xT}(i\,\Delta x\,, j\,\Delta T)\\
& + d(j\,\Delta T)\,\theta\, c_T(i\,\Delta x\,, j\,\Delta T)\Big) + \mathcal{O}((\Delta T)^2) + \mathcal{O}((\Delta x)^2)\,.
\end{aligned}$$

If the coefficients $b(T)\,, \hat{\sigma}(x,T)$ and $d(T)$ are not dependent on T, i. e.

$$b(T) = \bar{b}\,, \quad \hat{\sigma}(x,T) = \bar{\sigma}(x)\,, \quad d(T) = \bar{d}$$

then $\varepsilon_{i,j+1}$ becomes for $\theta = \frac{1}{2}$

$$\begin{aligned}
\varepsilon_{i,j+1} = \ & \tfrac{\Delta T}{2}\ \big(c_{TT}(i\,\Delta x\,, j\,\Delta T) + (\bar{b} + \tfrac{1}{2}\bar{\sigma}^2(i\,\Delta x))c_{xT}(i\,\Delta x\,, j\,\Delta T) + \bar{d}\, c_T(i\,\Delta x\,, j\,\Delta T)\\
& - \tfrac{1}{2}\bar{\sigma}^2(i\,\Delta x)\, c_{xxT}(i\,\Delta x\,, j\,\Delta T)\big) + \mathcal{O}((\Delta T)^2) + \mathcal{O}((\Delta x)^2)\,.
\end{aligned}$$

However, in this case $\varepsilon_{i,j+1} = \mathcal{O}((\Delta T)^2) + \mathcal{O}((\Delta x)^2)$ since $c(x,T)$ is a solution to the parabolic differential equation (5.2.12), i. e.

$$\begin{aligned}
& c_T(i\,\Delta x\,, j\,\Delta T) + (\bar{b} + \tfrac{1}{2}\bar{\sigma}^2(x_i))c_x(i\,\Delta x\,, j\,\Delta T) + \bar{d}\, c(i\,\Delta x\,, j\,\Delta T)\\
& \qquad - \tfrac{1}{2}\bar{\sigma}^2(x_i)\, c_{xx}(i\,\Delta x\,, j\,\Delta T) = 0
\end{aligned}$$

and therefore, the derivative with respect to T of the left hand side of the previous equation satisfies

$$\begin{aligned}
& c_{TT}(i\,\Delta x\,, j\,\Delta T) + (\bar{b} + \tfrac{1}{2}\bar{\sigma}^2(x_i))c_{xT}(i\,\Delta x\,, j\,\Delta T) + \bar{d}\, c_T(i\,\Delta x\,, j\,\Delta T)\\
& \qquad - \tfrac{1}{2}\bar{\sigma}^2(x_i)\, c_{xxT}(i\,\Delta x\,, j\,\Delta T) = 0
\end{aligned}$$

such that all the terms containing ΔT in the equation for $\varepsilon_{i,j+1}$ cancel out for all i and j. Thus, the Crank–Nicolson method is even consistent of order $(2,2)$ with respect to $\| \cdot \|_{2,\Delta x}$ if the coefficients $b\,,\hat{\sigma}$ and d in the parabolic differential equation (5.2.12) are independent of the time T. This concludes the proof of Theorem 5.4.10. ■

Bibliography

[1] L. B. G. Andersen and R. Brotherton-Ratcliffe. The equity option volatility smile: an implicit finite–difference approach. *The Journal of Computational Finance*, 1(2):5–37, 1998.

[2] J. Andreasen. Implied Modelling: Stable Implementation. Technical Report, Version November 1996, Department of Operations Research, University of Aarhus, Denmark, 1996.

[3] M. Avellaneda and P. Laurence. *Quantitative Modeling of Derivative Securities: From Theory to Practice*. Chapman & Hall/CRC, Boca Raton, Florida, USA, 2000.

[4] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, New York, 1974.

[5] S. Baun. Neuronale Netze in der Aktienkursprognose. In H. Rehkugler and H. G. Zimmermann, editors, *Neuronale Netze in der Ökonomie: Grundlagen und finanzwirtschaftliche Anwendungen*, pages 1–87. Vahlen, München, 1994.

[6] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications*. Robert E. Krieger Publishing Company, Huntington, New York, 1980.

[7] Å. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1996.

[8] Å. Björck, T. Elfving, and Z. Strakoš. Stability of conjugate gradient and Lanczos methods for linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 19(3):720–736, 1998.

[9] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[10] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Dissertation, Rheinische Friedrich–Wilhelms–Universität, Bonn, 1985.

[11] P. T. Boggs. The Convergence of the Ben-Israel Iteration for Nonlinear Least Squares Problems. *Mathematics of Computation*, 30(135):512–522, 1976.

[12] J. H. Bramble, A. H. Schatz, V. Thomée, and L. B. Wahlbin. Some Convergence Estimates for Semidiscrete Galerkin Type Approximations for Parabolic Equations. *SIAM Journal on Numerical Analysis*, 14(2):218–241, 1977.

[13] D. T. Breeden and R. H. Litzenberger. Prices of State–contingent Claims Implicit in Option Prices. *Journal of Business*, 51(4):621–651, 1978.

[14] K. Brockhoff. Prognosen. In F. X Bea et al., editors, *Allgemeine Betriebswirtschaftslehre*, volume 2: Führung, pages 551–590. Gustav Fischer Verlag, Stuttgart, 5th edition, 1991.

[15] K. M. Brown and J. E. Dennis. Derivative Free Analogues of the Levenberg–Marquardt and Gauss Algorithms for Nonlinear Least Squares Approximation. *Numerische Mathematik*, 18:289–297, 1972.

[16] Buchner. *Grundzüge der Finanzanalyse*. Vahlen, München, 1981.

[17] T. F. Coleman and A. Verma. Reconstructing the Unknown Local Volatility Function. *The Journal of Computational Finance*, 2(3), 1999.

[18] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust–Region Methods*. SIAM, MPS, Philadelphia, PA, 2000.

[19] H. Corsten and C. May. Anwendungsfelder Neuronaler Netze und ihre Umsetzung. In H. Corsten and C. May, editors, *Neuronale Netze in der Betriebswirtschaft*, pages 1–13. Gabler Verlag, Wiesbaden, 1996.

[20] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes*. Chapman and Hall, London, 1965.

[21] J. E. Dennis. Some Computational Techniques for the Nonlinear Least Squares Problem. In G. D. Byrne and C. A. Hall, editors, *Numerical Solution of Systems of Nonlinear Algebraic Equations*, pages 157–183. Academic Press, New York, 1973.

[22] J. E. Dennis. Non–linear Least Squares and Equations. In D. A. H. Jacobs, editor, *The State of the Art in Numerical Analysis*, pages 269–311. Academic Press, New York, 1977.

[23] J. E. Dennis. Algorithms for Nonlinear Fitting. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 67–78. Academic Press, New York, 1982.

[24] J. E. Dennis. A User's Guide to Nonlinear Optimization Algorithms. *Proceedings of the IEEE*, 72(12):1765–1776, 1984.

[25] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[26] J. E. Dennis and T. Steihaug. On the Successive Projections Approach to Least–Squares Problems. *SIAM Journal on Numerical Analysis*, 23(4):717–733, 1986.

[27] E. Derman and I. Kani. Riding on a Smile. *RISK*, 7(2):32–39, 1994.

[28] E. Derman, I. Kani, and N. Chriss. Implied Trinomial Trees of the Volatility Smile. *The Journal of Derivatives*, 3(4):7–22, 1996.

[29] P. Deuflhard and V. Apostolescu. A Study of the Gauss–Newton Algorithm for the Solution of Nonlinear Least Squares Problems. In D. Frehse, J., Pallaschke and U. Trottenberg, editors, *Special Topics of Applied Mathematics*, pages 129–150. North-Holland Publishing Company, 1980.

[30] P. Deuflhard and G. Heindl. Affine Invariant Convergence Theorems for Newton's Method and Extensions to Related Methods. *SIAM Journal on Numerical Analysis*, 16(1):1–10, 1979.

[31] P. DuChateau and D. Zachmann. *Applied Partial Differential Equations*. Harper & Row, New York, NY, 1989.

[32] B. Dupire. Pricing with a Smile. *RISK*, 7(1):18–20, 1994.

[33] B. Dupire. Pricing and Hedging with Smiles. In M. A. H. Dempster and S. R. Pliska, editors, *Mathematics of Derivative Securities*, pages 103–111. Cambridge University Press, Cambridge, 1997.

[34] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM Journal on Numerical Analysis*, 21(2):352–362, 1984.

[35] V. Faber and T. Manteuffel. Orthogonal error methods. *SIAM Journal on Numerical Analysis*, 24:170–187, 1987.

[36] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative Solution of Linear Systems. *Acta Numerica*, pages 57–100, 1992.

[37] A. Friedman. *Partial Differential Equations of Parabolic Type*. Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1964.

[38] D. M. Gay. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.

[39] N. A. Gershenfeld and A. S. Weigend. The Future of Time Series: Learning and Understanding. In N. A. Gershenfeld and A. S. Weigend, editors, *Times Series Prediction: Forecasting the Future and Understanding the Past*, pages 1–70. Addison–Wesley, 1993.

[40] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, New York, Toronto, Sydney, San Francisco, 1981.

[41] S. M. Goldfeld, R. E. Quandt, and H. F. Trotter. Maximization by Quadratic Hill–Climbing. *Econometrica*, 34(3):541–551, 1966.

[42] G. H. Golub and W. Kahan. Calculation the singular values and pseudoinverse of a matrix. *SIAM Journal on Numerical Analysis*, 2:205–224, 1965.

[43] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, second edition, 1989.

[44] C. W. J. Granger. Forecasting in Economics. In N. A. Gershenfeld and A. S. Weigend, editors, *Times Series Prediction: Forecasting the Future and Understanding the Past*, pages 529–538. Addison–Wesley, 1993.

[45] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, PA, 1997.

[46] A. Griewank. Tutorial on Computational Differentiation and Optimization. Technical Report, University of Michigan, 1994.

[47] L. Grippo. A Class of Unconstrained Minimization Methods for Neural Network Training. *Optimization Methods and Software*, 4:135–150, 1994.

[48] Ch. Großmann and H.-G. Roos. *Numerik partieller Differentialgleichungen*. B. G. Teubner, Stuttgart, 1992.

[49] B. Gustafsson, H.-O. Kreiss, and J. Oliger. *Time Dependent Problems and Difference Methods*. John Wiley & Sons, Inc, New York, New York, 1995.

[50] W.M. Häußler. A Local Convergence Analysis for the Gauss–Newton and Levenberg–Morrison–Marquardt Algorithms. *Computing*, 31:231–244, 1983.

[51] S. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, 1999.

[52] M. Heinkenschloß. *Gauss–Newton Methods for Infinite Dimensional Least Squares Problems with Norm Constraints*. Dissertation, Universität Trier, Trier, 1991.

[53] H. Henschel. *Wirtschaftsprognosen*. Vahlen, München, 1979.

[54] M. R. Hestenes. Pseudoinverses and conjugate gradients. *Communications of the ACM*, 18(1):40–43, 1975.

[55] M. R. Hestenes. *Conjugate Direction Methods in Optimization*. Springer–Verlag, New York, 1980.

[56] M. R. Hestenes and E. Stiefel. Methods of conjugate gradient for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[57] M. Huang and V. Thomée. Some Convergence Estimates for Semidiscrete Type Schemes for Time–Dependent Nonselfadjoint Parabolic Equations. *Mathematics of Computation*, 37(156):327–346, 1981.

[58] J. C. Hull. *Options, Futures, and Other Derivative Securities*. Prentice–Hall, Inc., Englewood Cliffs, 1993.

[59] J. C. Hull and A. White. The Pricing of Options on Assets with Stochastic Volatilities. *Journal of Finance*, 42(2):281–300, 1987.

[60] A. Irle. *Finanzmathematik: Die Bewertung von Derivaten*. B. G. Teubner, Stuttgart, 1998.

[61] Seizô Itô. *Diffusion Equations*. American Mathematical Society, 1992.

[62] L. W. Johnson and R. D. Riess. *Numerical Analysis*. Addison–Wesley, Reading, MA, second edition, 1982.

[63] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, New York, NY, 1991.

[64] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

[65] K. P. Kratzer. *Neuronale Netze: Grundlagen und Anwendungen*. Carl Hanser Verlag, München, Wien, 1993.

[66] O. A. Ladyženskaja, V. A. Solonnikov, and N. N. Ural'ceva. *Linear and Quasilinear Equatioins of Parabolic Type*. American Mathematical Society, Providence, Rhode Island, 1968.

[67] R. Lagnado and S. Osher. Reconciling Differences. *RISK*, 10(4):79–83, 1997.

[68] J. Y. Lequarré. Foreign Currency Dealing: A Brief Introduction. In N. A. Gershenfeld and A. S. Weigend, editors, *Times Series Prediction: Forecasting the Future and Understanding the Past*, pages 131–137. Addison–Wesley, 1993.

[69] K. Levenberg. A Method for the Solution of Certain Non–linear Problems in Least Squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[70] A. K. Louis. *Inverse und schlecht gestellte Probleme*. Teubner, Stuttgart, Germany, 1989.

[71] O. L. Mangasarian and M. V. Solodov. Serial and Parallel Backpropagation Convergence via Nonmonotone Perturbed Minimization. *Optimization Methods and Software*, 4:103–116, 1994.

[72] D. W. Marquardt. An Algorithm for Least–Squares Estimation of Nonlinear Parameters. *SIAM Journal Applied Mathematics*, 11(2):431–441, 1966.

[73] R. C. Merton. Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

[74] R. C. Merton. Option Pricing when Underlying Stock Returns are Discontinuous. *Journal of Financial Economics*, 3:124–144, 1976.

[75] J. J. Moré. The Levenberg–Marquardt Algorithm: Implementation and Theory. In G. A. Watson, editor, *Numerical Analysis, Proceedings of the Biennial Conference held at Dundee, June 28 - July 1, 1977*, pages 105–116. Springer–Verlag, Berlin, 1978.

[76] J. J. Moré. Recent Developments in Algorithms and Software for Trust Region Methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming. The State of the Art, Bonn 1982*, pages 258–287. Springer–Verlag, Berlin, 1983.

[77] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen. How fast are nonsymmetric matrix iterations. *SIAM Journal on Matrix Analysis and Applications*, 13(3):778–795, 1992.

[78] R. Neuneier and H. G. Zimmermann. How to Train Neural Networks. In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 373–423. Springer–Verlag, Berlin, Heidelberg, 1998.

[79] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[80] C. C. Paige and M. A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[81] E. E. Peters. *Chaos and Order in the Capital Markets*. John Wiley & Sons, Inc., New York, 1996.

[82] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach, London, 1970.

[83] M. J. D. Powell. A new algorithm for unconstrained optimization. In O. L. Mangasarian, K. Ritter, and J. B. Rosen, editors, *Nonlinear Programming*, pages 31–65. Academic Press, London, 1970.

[84] M. J. D. Powell. Convergence Properties of a Class of Minimization Algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, London, 1975.

[85] H. Ramsin and P.-Å Wedin. A Comparison of some Algorithms for the Nonlinear Least Squares Problem. *BIT*, 17:72–90, 1977.

[86] A.-P. Refenes, editor. *Neural Networks in the Capital Markets*, Chichester, 1995. John Wiley & Sons, Inc.

[87] H. Rehkugler and T. Poddig. Künstliche Neuronale Netze in der Finanzmarktanalyse: Eine neue Ära der Kursprognosen. In H. Corsten and C. May, editors, *Neuronale Netze in der Betriebswirtschaft*, pages 17–35. Gabler Verlag, Wiesbaden, 1996.

[88] M. Rubinstein. Implied Binomial Trees. *The Journal of Finance*, 69(3):771–818, 1994.

[89] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, Cambridge, 1986.

[90] S. Saarinen, R. B. Bramley, and G. Cybenko. Neural Networks, Backpropagation, and Automatic Differentiation. In A. Griewank and G. F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 31–42. SIAM, Philadelphia, 1992.

[91] E. W. Sachs and M. Schulze. Convergence of Gauss–Newton Methods for Underdetermined Systems Arising in Neural Networks. *Preprint, Universität Trier, Germany*, 1999.

[92] K. Sandmann. *Einführung in die Stochastik der Finanzmärkte*. Springer, Berlin, 2nd edition, 2001.

[93] R. Schaback. Convergence Analysis of the General Gauss–Newton Algorithm. *Numerische Mathematik*, 46:281–309, 1985.

[94] H. R. Schwarz. *Numerische Mathematik*. B. G. Teubner, Stuttgart, 1993.

[95] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, New York, 1989.

[96] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press, New York, 1985.

[97] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[98] T. Steihaug. The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[99] J. Stoer. *Numerische Mathematik 1*. Springer, Berlin, Heidelberg, 1994.

[100] R. Stöttner. *Finanzanalyse: Grundlagen der markttechnischen Analyse*. Oldenbourg, München/Wien, 1989.

[101] J. W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*. Springer, New York, 1995.

[102] S. W. Thomas. *Sequential Estimation Techniques for Quasi–Newton Algorithms*. Ph.D. dissertation, Res. Rep. TR 75-227, Dept. Computer Science, Cornell University, Ithaca, NY, 1975.

[103] V. Thomée. Finite Difference Methods for Linear Parabolic Equations. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of Numerical Aanalysis*, volume I, pages 5–196. North–Holland, Amsterdam, 1990.

[104] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. Springer, Berlin, 1997.

[105] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and their Uses*, pages 57–88. Academic Press, London, 1981.

[106] P.-Å. Wedin. Perturbation Theory for Pseudo–Inverses. *BIT*, 13:217–232, 1973.

[107] P. Wilmott, J. Dewynne, and S. Howison. *Option Pricing: Mathematical Models and Computation*. Oxford Financial Press, Oxford, 1996.

[108] J. Wloka. *Partielle Differentialgleichungen*. B. G. Teubner, Stuttgart, 1982.

[109] H. G. Zimmermann. Neuronale Netze als Entscheidungskalkül. In H. Rehkugler and H. G. Zimmermann, editors, *Neuronale Netze in der Ökonomie: Grundlagen und finanzwirtschaftliche Anwendungen*, pages 1–87. Vahlen, München, 1994.

Curriculum Vitae

Michaela Schulze

geboren am 16. November 1969
in Göttingen, Germany

08/1976 – 07/1980	Wilhelm–Busch–Grundschule, Ebergötzen
08/1980 – 07/1982	Orientierungsstufe Nord, Göttingen
08/1982 – 05/1989	**Abitur**, Theodor–Heuss–Gymnasium, Göttingen
09/1989 – 06/1991	**Ausbildung zur Bankkauffrau**, BHW Bausparkasse, Hameln
06/1991 – 09/1992	**Kreditsachbearbeiterin**, BHW Bausparkasse, Hameln
10/1992 – 04/1998	**Diplom–Wirtschaftsmathematikerin**, Studium der Wirtschaftsmathematik an der Universität Trier mit dem mathematischen Schwerpunkt Numerik und der betriebswirtschaftlichen Spezialisierung in Geld, Kredit, Währung, Finanzwirtschaft
05/1998 – 08/2002	**Wissenschaftliche Mitarbeiterin** im Fachbereich IV – Mathematik der Universität Trier am Lehrstuhl von Prof. Dr. E. W. Sachs
07/2000 – 12/2000	**Associate Researcher** an der Virginia Polytechnical Institute and State University, Blacksburg, Virginia, USA
08/2002	**Promotion zum Dr. rer. nat.** an der Universität Trier