Efficient Reflectance Calculations and Parameter Estimation Methods for Radiative Transfer Problems in Paper Industry Applications

Per Edström

FSCN
Fibre Science and Communication Network
- ett skogsindustriellt forskningsprogram vid Mittuniversitetet
EFFICIENT REFLECTANCE CALCULATIONS AND PARAMETER ESTIMATION METHODS FOR RADIATIVE TRANSFER PROBLEMS IN PAPER INDUSTRY APPLICATIONS

Per Edström

FSCN - Fibre Science and Communication Network,
Department of Engineering, Physics and Mathematics,
Mid Sweden University, SE-871 88 Härnösand, Sweden
ISSN 1650-5387 2007:40, Mid Sweden University

ABSTRACT

Very fast code for standardized d/0° reflectance calculations and an efficient Gauss-Newton parameter estimation method are introduced into the radiative transfer based light scattering simulation tool DORT2002, which makes it competitive in paper industry applications.

Outstanding problems in general radiative transfer theory are addressed in a paper industry application. The parameter estimation problem is given a least-squares formulation, different solution methods are evaluated, some characteristics of the problem are found, and the sensitivity of the solution is analyzed. Tests show that Gauss-Newton type methods are most suitable for the studied parameter estimation problem; superior performance was shown with respect to both robustness and speed. The parameter estimation problem is shown to be non-trivial and sometimes ill-conditioned.

The type of analyses made in this work give good insight in the character of the problem, and similar studies will be valuable in the future design of measurements and parameter estimation methods when using angle-resolved measurements to estimate also the asymmetry factor.

SAMMANDRAG

Mycket snabb kod för standardiserade d/0° reflektansberäkningar och en effektiv Gauss-Newton parameterestimeringsmetod introduceras i den radiative transfer baserade ljusspridnings-simuleringsverktyget DORT2002, vilket gör den konkurrenskraftig i tillämpningar inom pappersindustrin.

Den typ av analyser som genomförs i detta arbete ger god insikt i problemets karaktär, och liknande studier kommer att vara värdefulla i den framtida utvecklingen av mätningar och parameterestimeringsmetoder när vinkelupplösta mätningar skall användas för att estimera även asymmetrifaktorn.
# TABLE OF CONTENTS

**ABSTRACT** .......................................................................................................................... I

**SAMMANDRAG** .................................................................................................................... I

1. **INTRODUCTION** ......................................................................................................................... 1

2. **FASTER REFLECTANCE CALCULATIONS** .................................................................................. 1

3. **OPTIMIZATION METHODS FOR PARAMETER ESTIMATION** .................................................. 2

   3.1. **IMPLEMENTATIONS OF STANDARD OPTIMIZATION METHODS** ........................................... 3

   3.1.1. **Newton** ................................................................................................................................. 3

   3.1.2. **Quasi-Newton** ..................................................................................................................... 4

   3.1.3. **Truncated Newton** .............................................................................................................. 5

   3.1.4. **Gauss-Newton** .................................................................................................................... 5

   3.2. **MATLAB OPTIMIZATION FUNCTIONS** .................................................................................. 6

   3.2.1. **Simplex (Matlab: fminsearch)** .............................................................................................. 6

   3.2.2. **Quasi-Newton (Matlab Optimization Toolbox: fminunc)** ................................................... 7

   3.2.3. **Gauss-Newton (Matlab Optimization Toolbox: lsqnonlin)** ................................................... 7

   3.2.4. **SQP (Matlab Optimization Toolbox: fmincon)** ..................................................................... 7

   3.2.5. **Minimax (Matlab Optimization Toolbox: fminimax)** ........................................................... 7

   3.3. **KUBELKA-MUNK** ................................................................................................................... 7

4. **SENSITIVITY ANALYSIS** .......................................................................................................... 8

5. **PERFORMANCE TESTS AND RESULTS** .................................................................................. 9

   5.1. **EVALUATION OF FASTER REFLECTANCE CALCULATIONS** ............................................... 9

   5.2. **COMPARISON OF OPTIMIZATION METHODS FOR PARAMETER ESTIMATION** .................. 11

   5.3. **CHARACTERIZATION OF THE PARAMETER ESTIMATION PROBLEM** ................................. 13

       5.3.1. **Convexity of the Problem** ................................................................................................ 13

       5.3.2. **Sensitivity of the Solution** ................................................................................................ 15

6. **SUGGESTIONS FOR FUTURE WORK** ..................................................................................... 17

7. **DISCUSSION** ............................................................................................................................ 18

8. **ACKNOWLEDGEMENTS** ......................................................................................................... 19

**REFERENCES** ............................................................................................................................ 19
1. INTRODUCTION

DORT2002 [1] is a radiative transfer solution method adapted to light scattering in paper and print. It utilizes general radiative transfer theory to achieve better accuracy than the existing Kubelka-Munk method [2-4]. This is achieved through the angular resolution of the calculations and results. DORT2002 has been successively improved and evaluated [5], and has also been successfully applied to real problems [6-8].

For efficient use, it is necessary with fast calculations, and every effort has been made in this direction in the development of DORT2002. To compete with Kubelka-Munk, however, even higher speed is needed for pure reflectance calculations. This corresponds to the common usage of Kubelka-Munk, although it is only a part of what DORT2002 can do. One part of this report describes a specialized code for even faster reflectance calculations in DORT2002. Performance tests of the new code are made and comparisons with the earlier code and with Kubelka-Munk are illustrated.

Furthermore, inverse calculations (or parameter estimation) are essential, but have not yet been available in DORT2002. Calculating material parameters from reflectance measurements is straightforward in the simpler Kubelka-Munk method, but is an outstanding problem in general radiative transfer problems. In this report, this parameter estimation problem is formulated as a least-squares optimization problem, and a number of different standard optimization methods for its solution are implemented and evaluated.

The properties of this parameter estimation problem have not yet been fully investigated. In order to correctly interpret the results, some knowledge of the influence of measurement errors or noise on the parameters is needed. This knowledge can also be used for the design of experiments with minimal influence of measurement errors. The convexity of the problem is investigated and a perturbation analysis is performed, and the results of numerical experiments are discussed in the light of those findings.

Section 2 discusses the faster reflectance calculations, and section 3 goes through the optimization methods used in the inverse calculations. The sensitivity analysis is performed in section 4, and the test cases are presented in section 5 together with the results. Suggestions for future work are given in section 6, and section 7 gives a brief discussion of the findings.

2. FASTER REFLECTANCE CALCULATIONS

In the DORT2002 solution method [1], the azimuthally averaged intensity is given by the 0th Fourier component. Among the variables that depend only on the azimuthally averaged intensity are total reflectance, total transmittance, total absorptance and flux. There is also a method in DORT2002 that breaks the azimuthal loop after the first time instead of fulfilling the prescribed $2N$ times when such variables are all that is required. This gives a significant reduction in computation time [5].

In addition to this, even higher speed is needed for pure reflectance calculations in order to compete with Kubelka-Munk. To achieve this, a specialized code was developed that calculates only reflectance measures from material parameters, with all other DORT2002 parameters set to mimic the d/0° instrument geometry (which corresponds to the standardized usage of Kubelka-Munk in the paper industry [9-12]). This could readily be done as a shell to the older DORT2002, but to minimize computation time, a specialized code was developed. All unnecessary calculations were removed, recyclable calculations were not repeated, and spectral simulations were efficiently implemented by
3. OPTIMIZATION METHODS FOR PARAMETER ESTIMATION

There are a number of different reflectance quantities that can be measured. Measurements of reflectance factor, \( R \), are abundant in the paper industry. Standardized measurements of \( R \) use two measurements of the same medium but of different grammage, \( w \), and make it possible to determine the scattering coefficient, \( \sigma_s \), and the absorption coefficient, \( \sigma_a \), but not the asymmetry factor, \( g \). Angle-resolved measurements are needed in order to determine \( g \).

The parameter estimation problem is to find parameter values that minimize some distance measure between real measurements or design targets, and model predictions. In the current setting, using only the two reflectance measurements from standardized paper industry measurements, there are two scalar parameters to estimate, \( \sigma_s > 0 \) and \( \sigma_a > 0 \). A fixed value in \([-1, 1]\) is given to \( g \). This makes the parameter estimation a zero-residual problem. It could even be solved as a nonlinear equation without minimization. However, the current work is the first stage in a deeper study, where angle-resolved measurements will be used to determine also \( g \). This will result in an over-determined parameter estimation problem including noise. The current work aims at finding some characteristics of the problem, and to evaluate with which optimization methods to continue.

One way to introduce the distance measure to minimize is through an objective function that sums squared errors, such as

\[
F(x) = \frac{1}{2} \| f(x) \|_2^2 = \frac{1}{2} \sum_i f_i(x)^2 = \frac{1}{2} \sum_i \left( R_{\text{model}}(x, w_i) - R_{\text{measure}}(w_i) \right)^2.
\]  

(1)

This is statistically optimal if the measurement errors are normally distributed. In this work, \( R_{\text{model}}(x, w_i) \) are given by DORT2002 simulations, and \( R_{\text{measure}}(w_i) \) are given by measurements. Using only measurements of \( R \) for two grammages (which, together with the Kubelka-Munk model, is standardized) gives \( \sigma_s \) and \( \sigma_a \) well determined. If one defines the set of permissible parameter combinations as

\[
S = \{ (\sigma_s, \sigma_a) : \sigma_s > 0, \sigma_a > 0 \},
\]

(2)

one can state the parameter estimation problem in various ways, which makes it possible to use different optimization methods to find a solution. Of course, since the problem is constrained by

\[
x \in S,
\]

(3)

one needs to deal with this separately if unconstrained formulations are used. If the problem is convex and a suitable starting estimate is available, it should be possible to use an unconstrained formulation.

It should be pointed out that it is assumed throughout this work that \( R \) is given uniquely from the parameters. This is a quite reasonable assumption, especially for applied problems, but nevertheless it has never been proved in general. Indeed, not much is known at all regarding existence and uniqueness for the general radiative transfer problem.
In order to find a suitable optimization method for the DORT2002 parameter estimation problem, different optimization methods were implemented and tested. A comparison was made between implementations of different standard optimization methods, Matlab functions and Matlab Optimization Toolbox functions. The purpose of the comparison is two-fold. Firstly, the implementations of the standard optimization methods are error-tested, so that they give the same results as commercial solvers. Secondly, different kinds of optimization strategies are screened, to choose a limited number for a continued deeper study later. Preferably, only standard optimization methods will be chosen, provided that they perform sufficiently, with the possibility to include other methods if they stand out.

Results from the Kubelka-Munk model are included for comparison. It is a much simpler solution method for the radiative transfer problem, but it is fast and simple, and well established in industrial applications such as the pulp and paper, printing, textile and plastic industries. Its speed and ease of use explains much of its widespread use, and although its solutions are only approximate, the accuracy is sufficient in several practical applications. However, there are a number of problems and applications where higher accuracy is needed, which can be achieved with more general solutions methods for the radiative transfer problem like DORT2002. It is therefore relevant to compare such methods with Kubelka-Munk with respect to speed and accuracy.

The Kubelka-Munk model was also used to obtain starting estimates for the other optimization methods. There are at least two reasons for this. Firstly, the Kubelka-Munk model is simple enough to give closed formulas for the inverse calculations. Therefore no optimization is needed and the starting estimates are found fast. Secondly, the Kubelka-Munk model is the simplest special case of the class of so-called discrete ordinate solution methods for the radiative transfer problem. Since DORT2002 is a general discrete ordinate solution method, the Kubelka-Munk model provides a guaranteed feasible starting estimate at low cost. To find a feasible starting estimate at all can in some applications be a large problem in itself.

The purpose of this paper is not to develop the best possible method for parameter estimation in the radiative transfer problem; the field is so diverse that specialized routines are needed that exploit the special properties of each specific problem. Instead, the point is to study local and global properties when using standard optimization methods. The goal is also to make a choice of method based on stability and efficiency, without performing a theoretical analysis of why the respective method has those properties.

The different optimization methods, together with a corresponding problem formulation, are described briefly in separate sections below.

### 3.1. Implementations of Standard Optimization Methods

#### 3.1.1. Newton

With an unconstrained minimization formulation like

$$\min_x F(x),$$

(a classical approach is Newton's method. The typical Newton iteration consists of determining a search direction, \( p_k \), by solving

$$\nabla_x^2 F(x_k) p_k = -\nabla_x F(x_k),$$

and then updating the solution estimate through
\[ x_{k+1} = x_k + \alpha_k p_k. \] (6)

The step size parameter, \( \alpha_k \), is normally chosen in a line search procedure. The constant step size \( \alpha_k = 1 \) gives the pure form of Newton's method. Near a nonsingular minimum, the Hessian \( \nabla^2 x F(x_k) \) will be positive definite, and the convergence will be quadratic. The good convergence comes at the cost of expensive calculation of the Hessian. On the other hand, far from such a local minimum, the Hessian may be singular or the search direction may not be a decent direction because the Hessian is not positive definite. Thus, unless a very good starting estimate is available, the convergence may initially be very poor. The convergence also depends on the condition of the Hessian in the solution.

In this paper, a Newton method was implemented with a forward difference scheme for the gradient and the Hessian. The constraints (3) were kept by limiting \( p_k \) in (the rare) cases where \( x_k + p_k \notin S \).

Then an Armijo condition [13] was used for choosing step size in the line search. A good starting estimate was obtained from the Kubelka-Munk model. The algorithm was as follows.

\begin{verbatim}
x ← starting estimate from Kubelka-Munk
repeat until convergence or failure
    convergence and failure check
    G ← forward difference gradient
    H ← forward difference Hessian
    p ← solve Hp=-G, through factorization or Gaussian elimination
    α ← 1, then half α until Armijo condition fulfilled
    x ← x + αp
end
\end{verbatim}

From a purely numerical point of view, different finite difference intervals, \( h \), should be used for the gradient (about the square root of machine precision) and Hessian (a few decades larger) approximations for most accurate results [14]. This is used throughout the implementations in this work, and tests show that this indeed leads to better convergence properties.

### 3.1.2. Quasi-Newton

Newton's method computes the full Hessian in each iteration, which is expensive. An alternative is to continuously update an approximation to the Hessian, and thereby approximate the Newton direction. Quasi-Newton methods do this by building curvature information from the successive iterates \( x_k \) and the corresponding gradients \( \nabla x F(x_k) \) to formulate a quadratic model problem. The idea is to avoid the second derivative calculations, while still maintaining good convergence since the Hessian is progressively approached. There is a large number of Hessian updating methods, but the BFGS method [15-18] is considered to be the best for general purposes. Quasi-Newton methods have the advantage that they always give descent. A possible drawback, however, is that inaccurate derivative information may accumulate errors.

In this paper, a quasi-Newton method was implemented exactly as the Newton method in section 3.1.1, but with the Hessian approximated through a BFGS update. The algorithm was as follows.

\begin{verbatim}
x ← starting estimate from Kubelka-Munk
H ← forward difference Hessian at start
repeat until convergence or failure
    convergence and failure check
\end{verbatim}
\begin{verbatim}
G ← forward difference gradient
p ← solve Hp=-G, through factorization or Gaussian elimination
α ← 1, then half α until Armijo condition fulfilled
x ← x + αp
H ← BFGS update of Hessian approximation
end

3.1.3. Truncated Newton

Newton’s method solves Eq. (5) for the search direction \( p_k \) through a factorization or Gaussian elimination. An alternative is to use an iterative method to find an adequate approximation. This method is called truncated Newton, and may in some cases be much faster – especially for large problems – while almost maintaining the convergence properties. In the current study, however, the problem is not large, but the function evaluations are costly. Hence, truncated Newton is not expected to outperform the other methods.

In this paper, a truncated Newton method was implemented exactly as the Newton method in section 3.1.1, but a Conjugate Gradient (CG) method was used to solve for the search direction. The algorithm was as follows.

\begin{verbatim}
x ← starting estimate from Kubelka-Munk
repeat until convergence or failure
    convergence and failure check
    G ← forward difference gradient
    H ← forward difference Hessian
    p ← solve Hp=-G, through Conjugate Gradients
    α ← 1, then half α until Armijo condition fulfilled
    x ← x + αp
end

3.1.4. Gauss-Newton

With a least-squares formulation like

\[
\min_x \frac{1}{2} \Vert f(x) \Vert^2_2 = \min_x \frac{1}{2} \sum_i f_i(x)^2,
\]

a specialized method is Gauss-Newton [19]. The Gauss-Newton iteration is based on the idea to linearize \( f \) around the point \( x_k \) to obtain the linear least-squares problem

\[
\min_{p_k} \frac{1}{2} \Vert f(x_k) + J(x_k) p_k \Vert^2_2,
\]

where \( J \) is the Jacobian of \( f \). This is then solved for \( p_k \) by using for example a QR factorization.

Formally, although not normally used in solution methods, the search direction is found through the normal equations

\[
J(x_k)^T J(x_k) p_k = -J(x_k)^T f(x_k).
\]

The Gauss-Newton method can through this be compared with Newton’s method. The search direction equation (9) is actually the same as equation (5), where the gradient is given by

\[
G(x_k) = J(x_k)^T f(x_k)
\]
\end{verbatim}
and the Jacobian is used to approximate the Hessian through

$$ H(x_k) \approx J(x_k)^T J(x_k). $$

(11)

Gauss-Newton saves computations – by not computing the Hessian – possibly at the expense of decreased convergence rate. Near a solution the approximation is good and the convergence rate satisfactory; the convergence can be close to quadratic for zero-residual problems if \( J \) has full rank in the solution. The difference between the Hessian and its Gauss-Newton approximation can be shown to be \( \sum_i f_i(x) f_i^T(x) \), see equation (26). Hence, if the residual \( f_i(x) \) or curvature \( f_i^T(x) \) is large, or if the Jacobian is ill-conditioned in an iteration or in the solution, Gauss-Newton may converge slowly or not at all. In such cases the Newton method will be better.

It can be noted from equation (9) that if \( J \) is square and has full rank, as in the current study, the problem of finding the search direction can be reduced. In this case, the linearization of the problem (7) does not even need to be formulated as a least-squares problem (8), but can be reduced to solving – with zero residual – the linear system

$$ J(x_k)p_k = -f(x_k) $$

(12)

through Gauss elimination. However, since the current work is the first stage in a deeper study that will result in over-determined parameter estimation problems, the least-squares formulation is used here.

In this paper, a Gauss-Newton method was implemented with a forward difference scheme for the Jacobian, and an Armijo condition was used for choosing step size in the line search. The constraints (3) were kept by limiting \( p_k \) in (the rare) cases where \( x_k + p_k \notin S \), and the starting estimate was obtained from the Kubelka-Munk model. The algorithm was as follows.

\[
x \leftarrow \text{starting estimate from Kubelka-Munk}
\]

**repeat until convergence or failure**

  **convergence and failure check**
  \[
  J \leftarrow \text{forward difference Jacobian}
  \]
  \[
  p \leftarrow \text{solve} \min |f+Jp| \text{ through QR factorization}
  \]
  \[
  \alpha \leftarrow 1, \text{ then half } \alpha \text{ until Armijo condition fulfilled}
  \]
  \[
  x \leftarrow x + \alpha p
  \]

**end**

### 3.2. Matlab Optimization Functions

#### 3.2.1. Simplex (Matlab: fminsearch)

With an unconstrained minimization formulation like (4) it is possible to use derivative free direct search methods like the simplex method of Nelder and Mead [20]. Direct search methods do not use gradients, neither numerically nor analytically calculated, which is advantageous if they are expensive to calculate or approximate, or not even available. A simplex in \( n \)-dimensional space is an \( n+1 \) polyhedron, so in two-space a simplex is a triangle. At each step of the search, a new point is generated from the current simplex through operations like reflection, expansion and contraction. By comparison of function values at the vertices, the new point is included to replace one of the vertices, which gives a new simplex. This is repeated until the simplex is smaller than some tolerance. A simplex method is generally less efficient than other methods, but is instead generally more robust.

The Matlab function \texttt{fminsearch} was tested as an example of optimization methods of simplex type.
3.2.2. **Quasi-Newton (Matlab Optimization Toolbox: \texttt{fminunc})**

The function \texttt{fminunc} provided in Matlab Optimization Toolbox was tested as an example of unconstrained optimization methods of quasi-Newton type. It uses the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure.

3.2.3. **Gauss-Newton (Matlab Optimization Toolbox: \texttt{lsqnonlin})**

The function \texttt{lsqnonlin} provided in Matlab Optimization Toolbox was tested as an example of constrained optimization methods of Gauss-Newton type. It solves nonlinear least-squares problems using the Gauss-Newton method with a mixed quadratic and cubic line search procedure.

3.2.4. **SQP (Matlab Optimization Toolbox: \texttt{fmincon})**

Sequential quadratic programming (SQP) methods [21] are related to Newton’s method. At each iteration, the Hessian is approximated using a quasi-Newton updating method. This is then used to generate a quadratic programming (QP) subproblem whose solution is used to form a search direction for a line search procedure.

The function \texttt{fmincon} provided in Matlab Optimization Toolbox was tested as an example of constrained optimization methods of SQP type. It solves a quadratic programming (QP) subproblem at each iteration, and a positive definite quasi-Newton approximation of the Hessian is calculated using the BFGS method.

3.2.5. **Minimax (Matlab Optimization Toolbox: \texttt{fminimax})**

Minimax methods minimize the worst-case value of a set of multivariable functions,

\[
\min_{x} \max_{\{i\}} \{f_{i}(x)\},
\]

with an SQP method [22].

The function \texttt{fminimax} provided in Matlab Optimization Toolbox was tested as an example of optimization methods of minimax type. It uses a modified Hessian that takes advantage of the special structure of the minimax problem. In the current problem, however, errors will presumably be normally distributed with no outliers, and this makes the least-squares formulation optimal. Hence, the minimax formulation is not statistically motivated here, and \texttt{fminimax} is not expected to outperform the other methods.

3.3. **Kubelka-Munk**

The Kubelka-Munk model is simple enough to give closed formulas for the inverse calculations, so no optimization is needed. However, its parameters are not the physically objective scattering and absorption coefficients used in general radiative transfer theory. Therefore some translation is needed for comparison. No exact translation exists, but it is generally regarded that it is adequate to use the translation suggested by Mudgett and Richards [23-24], complemented with the compensation of anisotropic single scattering of van de Hulst [25], i.e.

\[
s = \frac{1}{4} \sigma_{s} (1 - g),
\]

\[
k = 2\sigma_{s}.
\]
4. SENSITIVITY ANALYSIS

To study the sensitivity for perturbation of the solution to the parameter estimation problem, some notation is needed. By defining

$$ F(x,b) = \frac{1}{2} \sum f_i(x,b)^2 = \frac{1}{2} \sum (M_i(x) - b_i)^2, $$

where $f_i(x,b)$ are the residuals, $M_i(x)$ the model predictions and $b_i$ the measurements, the nonlinear least-squares problem can be stated

$$ \min_x F(x,b), $$

with the solution $\hat{x}$, and the perturbed problem can be stated

$$ \min_{\delta x} F(\hat{x} + \delta x, b + \delta b), $$

with the solution $\hat{x} + \delta \hat{x}$. This can also be stated as

$$ \min_{\delta x} F(\hat{x} + \delta x, b + \delta b), $$

and a Taylor expansion around $(\hat{x}, b + \delta b)$ gives

$$ F(\hat{x} + \delta x, b + \delta b) = F(\hat{x}, b + \delta b) + \delta x^T \nabla_x F(\hat{x}, b + \delta b) + \frac{1}{2} \delta x^T \nabla_{xx} F(\hat{x}, b + \delta b) \delta x + O(\|\delta x\|^3). $$

The problem is now to find

$$ \min_{\delta x} g(\delta x), $$

which can be done by solving

$$ \nabla_{\delta x} g(\delta x) = 0, $$

which in turn is equivalent to solving

$$ \nabla_x F(\hat{x}, b + \delta b) + \nabla_{xx} F(\hat{x}, b + \delta b) \delta x = 0 $$

for $\delta x$. But by denoting by $J(\hat{x})$ the Jacobian of $f(x,b)$ evaluated in $\hat{x}$

$$ \nabla_x F(\hat{x}, b + \delta b) = J(\hat{x})^T f(\hat{x}, b + \delta b) = J(\hat{x})^T (f(\hat{x}, b) - \delta b) = -J(\hat{x})^T \delta b, $$

since the optimality conditions for the unperturbed problem gives

$$ J(\hat{x})^T f(\hat{x}, b) = 0. $$

In addition to this, it holds that

$$ \nabla_{xx}^2 F(\hat{x}, b + \delta b) = J(\hat{x})^T J(\hat{x}) + \sum_i f_i^T(\hat{x}) f_i(\hat{x}, b + \delta b_i), $$

where $f_i^T(\hat{x})$ is the Hessian of $f_i(x,b)$ evaluated in $\hat{x}$. Inserting this into equation (23) gives...
\[-J(\hat{x})^T \delta b + \left(J(\hat{x})^T J(\hat{x}) + \sum_i f_i(\hat{x})(f_i(\hat{x}, b) - \delta \vec{b})_j \right) \delta x = 0. \] (27)

Thus, the solution \(\hat{x} + \delta x\) to the perturbed problem (18) is approximately given by solving equation (27) for \(\delta x\). This gives the formal solution
\[
\delta x = \left(J(\hat{x})^T J(\hat{x}) + \sum_i f_i(\hat{x})(f_i(\hat{x}, b) - \delta \vec{b})_j \right)^{-1} J(\hat{x})^T \delta b
= \left(J(\hat{x})^T J(\hat{x}) + \sum_i f_i(\hat{x})(f_i(\hat{x}, b)) \right)^{-1} J(\hat{x})^T \delta b + O(\|\delta \vec{b}\|^2).
\] (28)

The relative change of parameter \(x_j\) for measurement perturbations is thus given by
\[
\frac{\delta x_j}{x_j} = \frac{1}{x_j} \sum_j P_{ij} \delta b_j + O(\|\delta \vec{b}\|^2) = \frac{1}{x_j} \sum_j P_{ij} b_j \delta b_j + O(\|\delta \vec{b}\|^2),
\] (29)

and so the relative sensitivity of parameter \(x_i\) for change in measurement \(b_j\) is approximately given by
\[
\kappa_{ij} = \frac{1}{x_j} P_{ij} b_j. \] (30)

Hence, if \(\kappa_{ij}\) is large, even small perturbations in measurement \(j\) will result in large changes in parameter \(i\). This may be the case for nonzero-residual problems with large curvature and for problems with large residual, but it may also happen for zero-residual problems with ill-conditioned Jacobian or Hessian in the solution.

5. PERFORMANCE TESTS AND RESULTS

To keep the tables and figures as clean as possible, no units are given there. Throughout this paper, scattering and absorption coefficients have the unit m²/kg, grammages have the unit g/m², and times have the unit s. Reflectance factors and asymmetry factors are dimensionless. Some test cases occur repeatedly, and are identified throughout by capital letters to ease comparisons.

5.1. Evaluation of Faster Reflectance Calculations

A set of test problems were solved with the Full DORT2002 code (going all 2N times through the azimuthal loop), the Partial DORT2002 code (going only once through the azimuthal loop), the new Specialized DORT2002 code, and with Kubelka-Munk. The results were compared with respect to speed and accuracy. The test problems and the results are found in tables 1-3.
Table 1. Test case B: \((\alpha_s = 14, \alpha_a = 5.6, g = 0, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>0.2098</td>
<td>0.2198</td>
<td>0.0766</td>
</tr>
<tr>
<td>Partial</td>
<td>0.2098</td>
<td>0.2198</td>
<td>0.0137</td>
</tr>
<tr>
<td>Specialized</td>
<td>0.2098</td>
<td>0.2198</td>
<td>0.0021</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.2526</td>
<td>0.2580</td>
<td>2.431e-05</td>
</tr>
</tbody>
</table>

Table 2. Test case C: \((\alpha_s = 14.7, \alpha_a = 0.03, g = 0, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>0.4394</td>
<td>0.8763</td>
<td>0.0722</td>
</tr>
<tr>
<td>Partial</td>
<td>0.4394</td>
<td>0.8763</td>
<td>0.0137</td>
</tr>
<tr>
<td>Specialized</td>
<td>0.4394</td>
<td>0.8763</td>
<td>0.0021</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.5217</td>
<td>0.9009</td>
<td>2.444e-05</td>
</tr>
</tbody>
</table>

Table 3. Test case E: \((\alpha_s = 84, \alpha_a = 5.0, g = 0.8, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>0.2113</td>
<td>0.2214</td>
<td>0.0932</td>
</tr>
<tr>
<td>Partial</td>
<td>0.2113</td>
<td>0.2214</td>
<td>0.0138</td>
</tr>
<tr>
<td>Specialized</td>
<td>0.2113</td>
<td>0.2214</td>
<td>0.0022</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.2981</td>
<td>0.3046</td>
<td>2.472e-05</td>
</tr>
</tbody>
</table>

As expected, the different DORT2002 codes all give the same accuracy, while Kubelka-Munk gives more approximate results (as discussed earlier by Edström [7]). As can be seen in figure 1, the Partial DORT2002 code is on average 6 times faster than the Full DORT2002 code, and the Specialized DORT2002 is about 50 times faster. Thus, the new Specialized DORT2002 code gives a significant reduction in computation time.

![Figure 1](image-url)  

Figure 1. The typical decrease in computation time between the Full DORT2002 code and the Specialized DORT2002 code is about a factor of 50, which is a significant reduction.
5.2. Comparison of Optimization Methods for Parameter Estimation

The different optimization methods were applied to a set of test problems. The methods were then compared with respect to speed and accuracy. The test problems and the results are found in the tables below.

As noted in section 3.1.1, different finite difference intervals should, from a purely numerical point of view, be used for the gradient and Hessian approximations for most accurate results [14]. This was initially not recognized, and the calculations using second derivatives explicitly – the Newton and truncated Newton methods and the sensitivity analysis – gave inconsequent results. Tests showed that using the square root of machine precision as finite difference interval for gradient approximations and a few decades larger for Hessian approximations lead to better convergence properties. This was then used throughout the implementations in this work.

From the results in tables 4-9, it can be noted that the optimization problem is not at all trivial. One case converged to a non-global minimum for fmincon, one case did not converge for Newton and truncated Newton, three cases lead to a singularity for quasi-Newton or truncated Newton, and in three cases did some unconstrained Matlab functions lead to an iterate $x_k \not\in S$. Thus, the nature of the problem demands from the optimization methods to handle this efficiently.

**Table 4.** Test case A: ($R_0^* = 0.42, R_\infty^* = 0.67, g = 0, w = 0.1$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$R_0$</th>
<th>$R_\infty$</th>
<th>$\sigma_i$</th>
<th>$\sigma_s$</th>
<th>Objective</th>
<th>Time</th>
<th>Iterations</th>
<th>Func.evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>0.41999</td>
<td>0.67</td>
<td>14.7600</td>
<td>0.30546</td>
<td>2.0402e-012</td>
<td>0.1456</td>
<td>5</td>
<td>43</td>
</tr>
<tr>
<td>quasi-Newton</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>2.6768e-014</td>
<td>0.1143</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>trunc. Newton</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>4.7426e-017</td>
<td>0.1541</td>
<td>5</td>
<td>44</td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>1.1648e-016</td>
<td>0.0383</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>1.5372e-017</td>
<td>0.2715</td>
<td>57</td>
<td>109</td>
</tr>
<tr>
<td>fminunc</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>5.1535e-014</td>
<td>0.1388</td>
<td>13</td>
<td>48</td>
</tr>
<tr>
<td>fmincon</td>
<td>0.42000</td>
<td>0.66999</td>
<td>14.7603</td>
<td>0.30547</td>
<td>3.2175e-012</td>
<td>0.0602</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>8.7934e-015</td>
<td>0.1962</td>
<td>19</td>
<td>60</td>
</tr>
<tr>
<td>fmincon</td>
<td>0.42</td>
<td>0.67</td>
<td>14.7602</td>
<td>0.30546</td>
<td>4.5133e-018</td>
<td>0.2947</td>
<td>20</td>
<td>104</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.34089</td>
<td>0.61817</td>
<td>10.6235</td>
<td>0.32376</td>
<td>0.00447</td>
<td>0.0024</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 5.** Test case B: ($R_0^* = 0.21, R_\infty^* = 0.22, g = 0, w = 0.1$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$R_0$</th>
<th>$R_\infty$</th>
<th>$\sigma_i$</th>
<th>$\sigma_s$</th>
<th>Objective</th>
<th>Time</th>
<th>Iterations</th>
<th>Func.evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>0.20997</td>
<td>0.22002</td>
<td>13.9901</td>
<td>5.58910</td>
<td>4.6862e-010</td>
<td>1.3559</td>
<td>23</td>
<td>463</td>
</tr>
<tr>
<td>quasi-Newton</td>
<td>0.21000</td>
<td>0.22</td>
<td>14.0087</td>
<td>5.59738</td>
<td>8.8701e-013</td>
<td>0.2011</td>
<td>16</td>
<td>77</td>
</tr>
<tr>
<td>trunc. Newton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Did not converge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.20999</td>
<td>0.22</td>
<td>14.0080</td>
<td>5.59710</td>
<td>1.0888e-014</td>
<td>0.0373</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.21</td>
<td>0.22</td>
<td>14.0080</td>
<td>5.59710</td>
<td>1.1264e-019</td>
<td>0.4006</td>
<td>90</td>
<td>175</td>
</tr>
<tr>
<td>fminunc</td>
<td>0.21</td>
<td>0.21999</td>
<td>14.0110</td>
<td>5.59883</td>
<td>1.0496e-010</td>
<td>0.1570</td>
<td>14</td>
<td>54</td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.20999</td>
<td>0.22000</td>
<td>14.0054</td>
<td>5.59595</td>
<td>2.0481e-011</td>
<td>0.0643</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>fmincon</td>
<td>0.20055</td>
<td>0.22686</td>
<td>10.0409</td>
<td>3.81456</td>
<td>6.8173e-005</td>
<td>0.0815</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>fminimax</td>
<td>0.20999</td>
<td>0.22002</td>
<td>13.9985</td>
<td>5.59249</td>
<td>1.9343e-010</td>
<td>0.2592</td>
<td>18</td>
<td>94</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.16981</td>
<td>0.18638</td>
<td>9.38242</td>
<td>4.86500</td>
<td>0.0014</td>
<td>0.0025</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table 6. Test case C: \((R_0^* = 0.44, R_\infty^* = 0.88, g = 0, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>(\sigma_s)</th>
<th>(\sigma_a)</th>
<th>Objective</th>
<th>Time (s)</th>
<th>Iterations</th>
<th>Func. evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>3.9438e-015</td>
<td>0.7482</td>
<td>20</td>
<td>284</td>
</tr>
<tr>
<td>quasi-Newton</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>8.3831e-019</td>
<td>0.0959</td>
<td>9</td>
<td>35</td>
</tr>
<tr>
<td>trunc. Newton</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>3.2678e-036</td>
<td>0.9960</td>
<td>24</td>
<td>360</td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>4.8130e-015</td>
<td>0.0356</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>2.4431e-017</td>
<td>0.2966</td>
<td>57</td>
<td>111</td>
</tr>
<tr>
<td>fminunc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.44000</td>
<td>0.87999</td>
<td>14.7247</td>
<td>0.02810</td>
<td>1.3776e-011</td>
<td>0.0598</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fmincon</td>
<td>0.44001</td>
<td>0.87998</td>
<td>14.7254</td>
<td>0.02811</td>
<td>3.1652e-010</td>
<td>0.1889</td>
<td>17</td>
<td>55</td>
</tr>
<tr>
<td>fminimax</td>
<td>0.44</td>
<td>0.88</td>
<td>14.7246</td>
<td>0.02810</td>
<td>5.802e-014</td>
<td>0.2264</td>
<td>14</td>
<td>76</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.35794</td>
<td>0.85173</td>
<td>10.5803</td>
<td>0.03246</td>
<td>0.00377</td>
<td>0.0027</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 7. Test case D: \((R_0^* = 0.42, R_\infty^* = 0.67, g = 0.8, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>(\sigma_s)</th>
<th>(\sigma_a)</th>
<th>Objective</th>
<th>Time (s)</th>
<th>Iterations</th>
<th>Func. evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>0.42001</td>
<td>0.66999</td>
<td>80.6102</td>
<td>0.29729</td>
<td>4.2290e-011</td>
<td>14.0188</td>
<td>355</td>
<td>5842</td>
</tr>
<tr>
<td>quasi-Newton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trunc. Newton</td>
<td>0.41999</td>
<td>0.67000</td>
<td>80.6023</td>
<td>0.29721</td>
<td>1.0369e-010</td>
<td>31.8363</td>
<td>719</td>
<td>13163</td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.42</td>
<td>0.67</td>
<td>80.6072</td>
<td>0.29727</td>
<td>7.4665e-014</td>
<td>0.0466</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.42</td>
<td>0.67</td>
<td>80.6071</td>
<td>0.29727</td>
<td>1.4746e-018</td>
<td>0.3313</td>
<td>67</td>
<td>126</td>
</tr>
<tr>
<td>fminunc</td>
<td>0.42</td>
<td>0.67</td>
<td>80.6070</td>
<td>0.29727</td>
<td>3.7835e-014</td>
<td>0.2322</td>
<td>18</td>
<td>63</td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.41999</td>
<td>0.67000</td>
<td>80.6066</td>
<td>0.29726</td>
<td>3.1009e-012</td>
<td>0.2209</td>
<td>21</td>
<td>67</td>
</tr>
<tr>
<td>fmincon</td>
<td>0.41999</td>
<td>0.67000</td>
<td>80.6066</td>
<td>0.29726</td>
<td>3.1009e-012</td>
<td>0.2209</td>
<td>21</td>
<td>67</td>
</tr>
<tr>
<td>fminimax</td>
<td>0.42</td>
<td>0.67</td>
<td>80.6071</td>
<td>0.29727</td>
<td>3.0051e-015</td>
<td>0.3275</td>
<td>22</td>
<td>115</td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.30835</td>
<td>0.59997</td>
<td>53.1175</td>
<td>0.32375</td>
<td>0.00868</td>
<td>0.0024</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 8. Test case E: \((R_0^* = 0.21, R_\infty^* = 0.22, g = 0.8, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_\infty)</th>
<th>(\sigma_s)</th>
<th>(\sigma_a)</th>
<th>Objective</th>
<th>Time (s)</th>
<th>Iterations</th>
<th>Func. evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>0.20988</td>
<td>0.22001</td>
<td>83.1915</td>
<td>4.99682</td>
<td>1.2075e-008</td>
<td>2.5279</td>
<td>58</td>
<td>1009</td>
</tr>
<tr>
<td>quasi-Newton</td>
<td>0.21001</td>
<td>0.21999</td>
<td>83.7219</td>
<td>5.03223</td>
<td>6.4586e-011</td>
<td>0.3536</td>
<td>20</td>
<td>133</td>
</tr>
<tr>
<td>trunc. Newton</td>
<td>0.21001</td>
<td>0.21999</td>
<td>83.7403</td>
<td>5.03346</td>
<td>1.4581e-010</td>
<td>4.2967</td>
<td>89</td>
<td>1696</td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.21</td>
<td>0.22</td>
<td>83.6853</td>
<td>5.02978</td>
<td>2.2660e-014</td>
<td>0.0366</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.21</td>
<td>0.22</td>
<td>83.6854</td>
<td>5.02979</td>
<td>4.4886e-021</td>
<td>0.3715</td>
<td>74</td>
<td>141</td>
</tr>
<tr>
<td>fminunc</td>
<td>0.21001</td>
<td>0.21999</td>
<td>83.7244</td>
<td>5.03235</td>
<td>7.5204e-011</td>
<td>0.2108</td>
<td>19</td>
<td>72</td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.20994</td>
<td>0.22004</td>
<td>83.4505</td>
<td>5.01432</td>
<td>5.4672e-009</td>
<td>0.0559</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>fmincon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fminimax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.13085</td>
<td>0.14592</td>
<td>46.9121</td>
<td>4.86500</td>
<td>0.00587</td>
<td>0.0025</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 9. Test case F: \((R_0^* = 0.44, R_a^* = 0.88, g = 0.8, w = 0.1)\).

<table>
<thead>
<tr>
<th>Method</th>
<th>(R_0)</th>
<th>(R_a)</th>
<th>(\alpha_2)</th>
<th>(\alpha_3)</th>
<th>Objective</th>
<th>Time</th>
<th>Iterations</th>
<th>Func.evals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Solution was not found after 1000 iterations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quasi-Newton</td>
<td></td>
<td></td>
<td>Did not converge - reached singularity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trunc. Newton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Solution was not found after 1000 iterations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gauss-Newton</td>
<td>0.44</td>
<td>0.88</td>
<td>0.02677</td>
<td>1.2130e-016</td>
<td>0.0520</td>
<td>9</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>fminsearch</td>
<td>0.44</td>
<td>0.88</td>
<td>0.02677</td>
<td>6.2638e-019</td>
<td>0.3057</td>
<td>63</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>fminunc</td>
<td></td>
<td></td>
<td>Unconstrained optimization went out of bounds ((\alpha_2 &lt; 0))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lsqnonlin</td>
<td>0.4400</td>
<td>0.87998</td>
<td>0.02679</td>
<td>3.1272e-010</td>
<td>0.0546</td>
<td>3</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>fmincon</td>
<td>0.43999</td>
<td>0.87999</td>
<td>0.02677</td>
<td>1.4196e-011</td>
<td>0.4052</td>
<td>21</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>fminimax</td>
<td>0.44</td>
<td>0.88</td>
<td>0.02677</td>
<td>1.5955e-016</td>
<td>0.4108</td>
<td>23</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>Kubelka-Munk</td>
<td>0.32664</td>
<td>0.84413</td>
<td>0.03246</td>
<td>0.00706</td>
<td>0.0025</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

As can be seen, apart from the simplex search method fminsearch, only the two Gauss-Newton methods – the standard implementation and the Matlab function lsqnonlin – converged for all test cases. In addition to this, the Gauss-Newton methods were by far fastest in all cases, and with good accuracy. This is also what could be expected, since, as noted in section 3.1.4, Gauss-Newton should be optimal for this problem setting. The standard Gauss-Newton method is on average almost twice as fast as the Matlab function lsqnonlin, probably because the Matlab functions are written to be general and safe.

The rest of the standard implementations (Newton, quasi-Newton and truncated Newton) performed reasonably well. They all converged in all but one case (but failed in different ones), and did so with good accuracy. As mentioned above, they were all considerably slower than Gauss-Newton, but there are still a few remarks to be made. That truncated Newton was slower was only to be expected, since it is designed for large problems while the current problem is small with costly function evaluations. However, it was not expected that Newton and quasi-Newton would be so much slower than Gauss-Newton. Kubelka-Munk gave, as expected, more approximate results (as discussed earlier by Edström [7]). By comparison with the results from the commercial solvers, it was found that all the implementations of standard optimization methods gave correct results upon convergence, and they were thereby error-tested.

As can be seen, there is clearly a difference in performance between the Matlab Optimization Toolbox function lsqnonlin and the other Matlab functions, the former being much faster. Surprisingly, the simplex search method fminsearch was about as fast as the other Matlab functions (except for lsqnonlin); one would rather expect it to be less efficient but more robust than the others. As expected, fminimax was slower than the other Matlab functions, since its minimax formulation is not statistically motivated here. All the Matlab functions seem to have good performance – when they converge – when it comes to accuracy in the final result.

5.3. Characterization of the Parameter Estimation Problem

In order to characterize the parameter estimation problem itself, not including any methods to solve it, some investigations were done to illustrate the convexity of the problem and the sensitivity of the solution.

5.3.1. Convexity of the Problem

The convexity of the problem was studied by plotting the objective function \(F(x)\) as function of the scattering and absorption parameters for three different test cases. A well-behaved problem has a
smooth and convex surface with one distinct minimum corresponding to the solution. Typical reasons for ill-conditioning include lack of smoothness or convexity or the existence of several local minima, which makes a global minimum hard to find, but also a very flat surface, which gives poor convergence rate and a sensitive solution.

Figure 3 indicates that cases with low opacity give well-conditioned problems with non-sensitive solutions. This is seen since the objective function surface is smooth and convex, and there is one distinct global minimum. Cases with high opacity (figures 2 and 4), on the other hand, seem to have an objective function surface that is flat in one or more directions, which shows that those cases give ill-conditioned problems with poor convergence (hard to find iteration steps in the optimization that give sufficient descent in the flat areas) or sensitive solutions (a small change in target value can give a large change in the parameter solution). This was also numerically investigated by changing $R_0$ with 0.1%, and it was noted that the relative change in the parameter solution was a factor 7 larger in the cases from figures 2 and 4 compared to the case from figure 3.

---

**Figure 2.** Test case B: ($\sigma_s = 14$, $\sigma_a = 5.6$, $g = 0$, $w = 0.1$), corresponding to an opacity of 95.5%. The objective function surface is smooth and convex, but is locally flat along a line, possibly giving poor convergence or a sensitive solution. The diamond indicates the point of convergence.

**Figure 3.** Test case C: ($\sigma_s = 14.7$, $\sigma_a = 0.03$, $g = 0$, $w = 0.1$), corresponding to an opacity of 50.1%. The objective function surface is smooth and convex with one distinct local minimum. The problem should be well conditioned with a non-sensitive solution. The solution is, however, close to a boundary, which may give rise to problems. The diamond indicates the point of convergence.
It is expected from application experience that cases with high opacity should give more ill-conditioned problems. In the standardized paper industry measurements used here, high opacity means that the two reflectance measurements are very similar and in the end indistinguishable. This will of course give rise to sensitivity, and it is well known in the paper industry that parameter estimation is problematic for samples with high opacity. It should be emphasized that the ill-conditioning is not introduced by any simulation model used, but is inherent in the problem.

The conditioning of the problem also depends on the asymmetry factor, $g$, since higher $g$ also flattens the objective function surface. The principle difference can be seen between figures 2 and 4, which differ primarily in the asymmetry factor. The case with higher $g$ (figure 4) is much flatter.

This all agrees with the findings in the previous section that the parameter estimation problem is not at all trivial. There, convergence to a non-global minimum, lack of convergence, singularities and iterates $x_k \notin S$ were encountered. It is seen from the figures that the curvature of the problem may in some cases be far from the quadratic approximations used locally by the optimization methods, and the minimum sometimes lies very close to the border of $S$.

### 5.3.2. Sensitivity of the Solution

The sensitivity of the solution was studied by generating contour plots of how $\sigma_s$ and $\sigma_a$ depend on $R_s$ and $R_o$ for two different values of the asymmetry factor $g$. Areas where the contour lines lie close are more sensitive, since a small change in a target value will there give a large change in the parameter solution. Studying these phase space plots gives visual information on which parameters are sensitive to what measurements, and how much in different areas. The sensitivity of the solution was also studied numerically, by calculating the sensitivity matrix $\kappa$ introduced in equation (30) for three different cases. The elements $\kappa_{ij}$ give quantitative information on the relative sensitivity of parameter $x_i$ for change in measurement $b_j$, where $x = (\sigma_s, \sigma_a)^T$ and $b = (R_s, R_o)^T$ in the current setting.

Both the phase space plots and the sensitivity matrix can be used to interpret the results from the parameter estimation, by giving knowledge on the influence of measurement errors or noise on the
parameters. They can also be used to design better experiments and measurements by devising measurements with minimal influence of noise and errors. It should be pointed out that the sensitivity is not introduced or affected by any simulation model used, but is a property of the problem itself.

It can be seen from the right hand panes of figure 5 that the scattering coefficient \( \sigma_s \) increases rapidly with \( R_o \) but decreases slightly with \( R_w \), and that the rate of change is larger for strongly absorbing samples, i.e. the contour lines lie closer in the lower left part of the figure. This means that a small error in \( R_o \) can cause large deviations in \( \sigma_s \), and that the relative size of the deviation is larger for highly absorbing samples. It is also obvious that \( \sigma_s \) is highly sensitive to measurement errors in regions close to the line \( R_o = R_w \), since the contour lines lie very close in this region. This once again illustrates the sensitivity in cases with high opacity. The absorption coefficient \( \sigma_a \) shows a similar dependence on the reflectances, i.e. it increases with \( R_o \) and decreases with \( R_w \), and it is highly sensitive to measurement errors for strongly absorbing samples and in regions close to the line \( R_o = R_w \).

![Figure 5](image-url)

**Figure 5.** Test cases A-C \((g = 0, w = 0.1)\) in upper panes and cases D-F \((g = 0.8, w = 0.1)\) in lower panes. Contour plot showing how \( \sigma_s \) (left) and \( \sigma_a \) (right) depend on \( R_o \) and \( R_w \). Darker color corresponds to higher value. The parameters increase rapidly with \( R_o \), but decrease slightly with \( R_w \). The problem is highly sensitive in regions near the line \( R_o = R_w \), i.e. in cases with very high opacity.
The relative sensitivities given in table 10 show, for example, that \( \sigma_y \) is about 5 times more sensitive in cases B and E (opacity around 95%) than in case C (opacity around 50%). Contrary to this reasoning, it is seen that the relative sensitivity of \( \sigma_y \) for change in \( R_a \) is much larger in case C. To explain this, the sensitivity matrix \( \kappa \) can be analyzed in detail from equations (28). It is evident that the sensitivity depends on the condition given by \( J^TJ \), the curvature given by the Hessian \( H \), and the residual \( f \). Tests show that the residual in the solution is so small in the tested cases, that \( J^TJ \) dominates. However, in some cases the curvature is larger, as for \( \sigma_a \) in case C (see figure 3), which then shows as a higher sensitivity.

### Table 10. Relative sensitivity of parameter \( i \) (1 = \( \sigma_y \), 2 = \( \sigma_a \)) for change in measurement \( j \) (1 = \( R_b \), 2 = \( R_a \)) in test cases B, C and E (from left to right).

<table>
<thead>
<tr>
<th>Test data</th>
<th>( R_b = 0.21, R_a = 0.22 )</th>
<th>( R_b = 0.44, R_a = 0.88 )</th>
<th>( R_b = 0.21, R_a = 0.22 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \kappa )</td>
<td>( g=0, w=0.1 )</td>
<td>( g=0, w=0.1 )</td>
<td>( g=0.8, w=0.1 )</td>
</tr>
<tr>
<td>( \kappa_{yx} )</td>
<td>6.7941 -5.8564</td>
<td>1.7347 -0.1408</td>
<td>6.2974 -5.5234</td>
</tr>
<tr>
<td>( \kappa_{xy} )</td>
<td>6.7941 -7.4877</td>
<td>1.7332 -16.5015</td>
<td>6.3093 -6.9964</td>
</tr>
</tbody>
</table>

### 6. SUGGESTIONS FOR FUTURE WORK

The parameter estimation problem will be studied further. This includes the continued study and development of fast and numerically stable algorithms for parameter estimation in the radiative transfer equation. The problem involves several numerical difficulties, so it will require some effort. Inverse solution methods for angle-resolved intensity measurements, including the ability to estimate the asymmetry factor \( g \), will be developed and evaluated. The parameter estimation will be done to fit model simulations to angle-resolved light scattering measurements or to desired angle-resolved light scattering patterns. Also, studies of error estimation and sensitivity of perturbations will be performed.

As can be seen in figures 2 and 4, high opacity cases seem to have an objective function surface that is locally flat along a line through the solution. If it were possible to identify this line a priori, the convergence could be accelerated. It would then be possible to develop tailor made methods with very good convergence properties. Although this might be interesting, this is not planned at the moment.

The simulation model will be adapted to include the influence of real, non-ideal, instruments in the interpretation of measurement data, in order to make it possible to estimate parameters more closely to the actual physical quantity. This would make different instruments – e.g. the d/0° geometry used in the paper industry and the 45°/0° geometry used in the graphical industry – more comparable, and would give objective and exchangeable parameters. This would provide a link between the paper and graphical industries, and explain the different ranking of samples.

The possibility to estimate objective physical parameters – instead of model and geometry dependent ones – may give more accurate determination of ink parameters, and may for example resolve the problem with different ink parameter values depending on the substrate used.
When the asymmetry factor \( g \) is possible to estimate, the possibility arises to study new material phenomena such as different direction dependencies in reflectance and transmittance. This opens up for understanding, but also for the design of new products.

7. DISCUSSION

The Kubelka-Munk model is a simple solution method for the radiative transfer problem, but is well established in several industrial applications due to its speed and ease of use. Its approximate solutions are sufficiently accurate in several practical applications, but there are a number of problems and applications where higher accuracy is needed. This can be achieved with more general solution methods for the radiative transfer problem, like DORT2002. To compete with Kubelka-Munk in industrial applications, higher accuracy is not enough. Sufficient speed is essential, as well as fast and accurate parameter estimation methods. The specialized code for standardized \( d/0^\circ \) reflectance calculations and the efficient Gauss-Newton parameter estimation method reported in this work show that radiative transfer based solution methods like DORT2002 are now competitive in paper industry applications.

Calculating material parameters from reflectance measurements is an outstanding problem in general radiative transfer problems. The formulation of the parameter estimation problem as a least-squares optimization problem, and the solution method investigation is a contribution to this field. With its higher accuracy, larger range of applicability and comparable speed, DORT2002 could well replace Kubelka-Munk in, for example, the paper industry for increased understanding. The Kubelka-Munk model will still be useful, though, in applications with lower demands on accuracy, and for providing a guaranteed feasible starting estimate in inverse DORT2002 calculations at low cost. To find a feasible starting estimate at all can in some applications be a large problem in itself.

One purpose of this work was to evaluate with which optimization methods to continue the future work. The results of this study are very clear. The tests show that the Gauss-Newton type methods are the most suitable optimization methods for the studied parameter estimation problem; superior performance was shown with respect to both robustness and speed. Gauss-Newton was the only method – apart from the slower simplex search method – to converge for all test cases. In addition to this, it was by far the fastest in all test cases, and it also turned out that the standard implementation was as accurate as commercial solvers, but twice as fast (probably since commercial solvers are written to be general and safe). It is therefore clear that Gauss-Newton is selected for future work. However, it will be complemented with Newton for reasons given below.

The optimization problem in this work is a zero-residual problem. It might therefore be expected that the Newton, truncated Newton, quasi-Newton and Gauss-Newton methods would have similar performance. It is not clear why the Newton methods did not compete better. Gauss-Newton saves computations by not computing the Hessian, but Newton seems to need more iterations in spite of its second order information. It is known that Gauss-Newton rejects saddle points, but it is not clear if the Newton methods have had such problems here. Also, Gauss-Newton is independent of scaling in the parameters while Newton is not, but here the parameters are rather equally scaled. However, there is always a dependence on the “shape” of the objective function of the specific problem, and on the starting estimate. Gauss-Newton may have a larger attraction area than Newton for some problems. In addition, Gauss-Newton always gives a descent direction in every step, which is not always the case for Newton. This may explain some of the performance differences. The exact implementation of the methods also has an influence. An extended work is planned, where angle-resolved intensity
measures will be used as target. That problem will also include measurement noise, and will not be a zero-residual problem. It is expected that the Newton method will outperform the other methods in that case.

Another purpose of this work was to find some characteristics of the studied parameter estimation problem, since not much is known at all regarding existence and uniqueness for the general radiative transfer problem and its inverse. The investigations show that the optimization problem is not at all trivial, and the nature of the sometimes ill-conditioned problem thus demands a lot from the optimization methods. The relative sensitivities found in the investigations are low, however, and should not be a problem in paper industry applications. The type of analyses made in this work – using objective function surface plots, phase space plots and sensitivity matrices – give good insight in the character of the problem, and similar studies will be valuable in the future design of measurements and parameter estimation methods when using angle-resolved measurements to estimate also the asymmetry factor.

8. ACKNOWLEDGEMENTS

This report was financially supported by T2F (“TryckTeknisk Forskning, a Swedish printing research program), which is gratefully acknowledged.

REFERENCES


