Chapter 6

Derivative Calculation

In the previous Chapter 5 we have introduced various iterative approaches to the nonlinear inverse transport problem. These methods were based on minimizing an objective function, which described the discrepancy between the measured and predicted data. All methods require for this minimization the first derivative of the objective function with respect to the unknown optical parameters. However, calculating this derivative is a quite difficult task itself, because the objective function depends on a large number of unknown optical parameters. Thus, the challenge is to find a fast and computationally efficient way of calculating this derivative.

A straightforward approach would be to approximate the derivative with divided differences:

$$\nabla_{\boldsymbol{\mu}} \Phi \approx \frac{\Delta \Phi}{\Delta \boldsymbol{\mu}} = \frac{\Phi(\boldsymbol{\mu} + \Delta \boldsymbol{\mu}) - \Phi(\boldsymbol{\mu})}{\Delta \boldsymbol{\mu}} \tag{6.1}$$

If $\boldsymbol{\mu}$ is a vector of N unknown optical properties one has to run $\mathrm{N}+1$ forward problems to obtain the gradient. Given that in OT the number of unknowns is typically $10^2 - 10^5$, this requirement results in an unacceptable computational burden [1].

[1] An example with $\mathrm{N} = 2 \times \mathrm{I} \times \mathrm{J} = 2 \times 60 \times 60 = 7,200$ unknown optical parameters requires 7,201

Instead of perturbing each component of the vector $\boldsymbol{\mu}$ using Equation 6.1, we have developed an *adjoint differentiation* scheme to calculate the derivative of the objective function. This approach needs approximately as many numerical operations for calculating the derivative as the forward model needs for solving one forward problem. The adjoint differentiation technique is N times faster than the method of divided differences, and is therefore a very powerful tool for evaluating derivatives of functions with many unknown variables.

6.1 Adjoint Model

The adjoint differentiation technique originates from the concept of *adjoint models*. An adjoint model is a tool developed for inverse modeling of physical systems. It determines the derivative of some quantity of the physical system with respect to given input parameters. The adjoint model is increasingly used in meteorology and oceanography for sensitivity studies, data assimilation, and parameter estimation [Hall82] [Hall86] [Navon97] [Giering00]. Errico gave a concise overview of adjoint models in atmospherical sciences for sensitivity studies [Errico97].

To understand the concept of the adjoint model it is necessary to introduce the *tangent linear model* that can be understood in the following manner. A forward model B, for example an ocean circulation model or climate model, maps a vector of *input* parameters $\boldsymbol{a}$ onto a vector of *output* parameters $\boldsymbol{b}$ with

$$\boldsymbol{b} = B(\boldsymbol{a}). \tag{6.2}$$

The input parameters $\boldsymbol{a}$ are usually the unknown model parameters, whereas the output

function evaluations $\Phi(\mu)$ for calculating the gradient with divided differences. Assuming that each function evaluation requires 1 minute using a PENTIUM III XEON® processor, we need approximately 5 days for calculating the derivative.

parameters $\boldsymbol{b}$ are the model predictions. Furthermore, the linearization of the forward model B maps variations $\delta\boldsymbol{a}$ of the input variables onto variations $\delta\boldsymbol{b}$ of the model predictions

$$\delta\boldsymbol{b} = \frac{\partial B}{\partial \boldsymbol{a}}\delta\boldsymbol{a}. \tag{6.3}$$

Equation 6.3 is also called the tangent linear model. It obtains information about the model predictions $\boldsymbol{b}$ from the input parameters $\boldsymbol{a}$.

The transition from the tangent linear model to the adjoint model is made by introducing the forecast error or residual R. The forecast error R measures the difference between the model predictions $\boldsymbol{b}$ and the measured data. Variations δR of the forecast error are derived from variations $\delta\boldsymbol{b}$ of the model predictions:

$$\delta R = \frac{\partial R}{\partial \boldsymbol{b}}\delta\boldsymbol{b}. \tag{6.4}$$

The adjoint model, in turn, provides the sensitivity $\partial R/\partial\boldsymbol{a}$ of the forecast error R with respect to the unknown model parameters $\boldsymbol{a}$ [Talagrand91a]. In contrast to the tangent linear model, which maps variations $\delta\boldsymbol{a}$ of the input parameters onto variations $\delta\boldsymbol{b}$ of the model predictions, the adjoint model infers information about the input variables $\boldsymbol{a}$ from the model predictions $\boldsymbol{b}$.

The derivative $\partial R/\partial\boldsymbol{a}$ can be derived in the following way. Using the inner product notation $\langle\cdot,\cdot\rangle$, the first-order variation δR (see Equation 6.4) resulting from a perturbation $\delta\boldsymbol{b}$ can also be written as

$$\delta R = \left\langle \frac{\partial R}{\partial \boldsymbol{b}}, \delta\boldsymbol{b} \right\rangle. \tag{6.5}$$

By using the tangent linear model (see Equation 6.3) we replace $\delta\boldsymbol{b}$ and get

$$\delta R = \left\langle \frac{\partial R}{\partial \boldsymbol{b}}, \frac{\partial B}{\partial \boldsymbol{a}}\delta\boldsymbol{a} \right\rangle. \tag{6.6}$$

We obtain from Equation 6.6 by using the definition $\langle \boldsymbol{q}, \boldsymbol{A}\boldsymbol{r}\rangle = \langle \boldsymbol{A}^* \boldsymbol{q}, \boldsymbol{r}\rangle$ for adjoint matrices $\boldsymbol{A}^*$ and vectors $\boldsymbol{q}$ and $\boldsymbol{r}$:

$$\delta R = \Big\langle \left(\frac{\partial B}{\partial \boldsymbol{a}}\right)^* \frac{\partial R}{\partial \boldsymbol{b}}, \delta \boldsymbol{a} \Big\rangle. \tag{6.7}$$

Comparing Equation 6.7 with the identity

$$\delta R = \Big\langle \frac{\partial R}{\partial \boldsymbol{a}}, \delta \boldsymbol{a} \Big\rangle, \tag{6.8}$$

the first term within the brackets of Equation 6.7 constitutes the gradient $\partial R / \partial \boldsymbol{a}$:

$$\frac{\partial R}{\partial \boldsymbol{a}} = \left(\frac{\partial B}{\partial \boldsymbol{a}}\right)^* \frac{\partial R}{\partial \boldsymbol{b}}. \tag{6.9}$$

Equation 6.9 represents the adjoint model. Further details can be found in [Talagrand91a].

There are three important differences between the tangent linear model (Equation 6.3) and the adjoint model (Equation 6.9). First, Equation 6.3 relates perturbations but Equation 6.9 relates derivatives. Second, Equation 6.3 takes perturbations of the input parameters to determine perturbations of the output parameters, but in Equation 6.9 the roles of input and output are reversed. Third, in Equation 6.3 input and output parameters are connected by the Jacobian $\frac{\partial B}{\partial \boldsymbol{a}}$ but in Equation 6.9 they are connected by its adjoint $\frac{\partial B}{\partial \boldsymbol{a}}^*$.

Now we will derive the corresponding formulation of the adjoint model for OT by using the modalities, just described, from meteorology and oceanography. In OT the input variable of the forward model is the N-dimensional vector $\boldsymbol{\mu}$ of optical parameters. Using this input parameter the transport forward model calculates the model predictions $\boldsymbol{p}$ at the specified source-detector positions. A linearization of the transport forward model F

around a point $\boldsymbol{\mu}_0$ is represented by the Jacobian matrix $\mathcal{J}(\boldsymbol{\mu}_0)$:

$$\mathcal{J} = \frac{\partial F}{\partial \boldsymbol{\mu}} = \begin{pmatrix} \frac{\partial F_1}{\partial \mu_1} & \frac{\partial F_2}{\partial \mu_1} & \cdots \\ \frac{\partial F_1}{\partial \mu_2} & \frac{\partial F_2}{\partial \mu_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

The tangent linear model maps variations $\delta\boldsymbol{\mu}$ in the optical parameters onto variations $\delta\boldsymbol{p}$ of the predictions at specified source-detector pairs by using the Jacobian matrix:

$$\delta\boldsymbol{p} = \mathcal{J}\delta\boldsymbol{\mu}. \tag{6.10}$$

At this point it is interesting to note that the perturbation technique in OT with $\Delta\boldsymbol{m} = \mathcal{J}\Delta\boldsymbol{\mu}$ (see Chapter 1.2.2), is equivalent to the tangent linear model. Differences $\Delta\boldsymbol{m}$ in the measurements are connected to differences $\Delta\boldsymbol{\mu}$ in the optical parameters by the Jacobian matrix $\mathcal{J}$. This matrix is very large [2] and leads to a huge computational burden when $\mathcal{J}$ is inverted to determine an update $\Delta\boldsymbol{\mu}$ of the optical parameters.

In contrast to the tangent linear model in OT, the adjoint model associates the influence of the optical parameters $\boldsymbol{\mu}$ on a given disparity of the measurements $\boldsymbol{m}$ and detector predictions $\boldsymbol{p}$. This residual is represented by the derivative $\nabla_{\boldsymbol{p}}\Phi \sim (\boldsymbol{p} - \boldsymbol{m})$ (see Equation 5.1), whereas the influence of the optical parameters on the objective function is the gradient $\nabla_{\mu}\Phi$. The derivation of the adjoint model can be performed in the following manner. The variation of the objective function, given the variations $\delta\boldsymbol{p}$ of the predicted detector readings, is

$$\delta\Phi = \langle \nabla_{\boldsymbol{p}}\Phi, \delta\boldsymbol{p} \rangle. \tag{6.11}$$

Substituting Equation 6.10 into Equation 6.11 yields

$$\delta\Phi = \langle \nabla_{\boldsymbol{p}}\Phi, \mathcal{J}\delta\boldsymbol{\mu} \rangle. \tag{6.12}$$

[2] For example, a problem with $\mathrm{N} = 2 \times \mathrm{I} \times \mathrm{J} = 2 \times 60 \times 60 = 7,200$ unknown optical parameters and $\mathrm{D} = 10^3$ source-detector pairs leads to a matrix $\mathcal{J}$ with $7.2 \cdot 10^6$ entries.

Using the definition $\langle v, Tu \rangle = \langle T^* v, w \rangle$ of the adjoint operator T^* we obtain:

$$\delta\Phi = \langle \mathcal{J}^* \nabla_{\boldsymbol{p}} \Phi, \delta\boldsymbol{\mu} \rangle. \tag{6.13}$$

Comparing Equation 6.13 with the identity

$$\delta\Phi = \langle \nabla_{\boldsymbol{\mu}} \Phi, \delta\boldsymbol{\mu} \rangle, \tag{6.14}$$

the gradient $\nabla_{\boldsymbol{\mu}} \Phi$ of the objective function is calculated as:

$$\nabla_{\boldsymbol{\mu}} \Phi = \mathcal{J}^* \nabla_{\boldsymbol{p}} \Phi. \tag{6.15}$$

Equation 6.15 represents the adjoint model in OT.

6.2 Numerical Implementation of the Adjoint Model

The adjoint model is represented by a system of differential equations (see Equation 6.15). In general, a system of equations can be solved numerically in three steps. First, the continuous differential equations are formulated. Second, a discretization scheme is chosen and the discrete difference equations are constructed. The last step is to implement an algorithm that solves the discretized equations. The adjoint model can be constructed after completion of any one of these three steps [Kaminski99].

In the first approach one obtains the gradient by using the solution of the adjoint ERT. The challenge in this approach is to derive the adjoint equation from a given forward model and to solve it. A general overview on the theory of adjoint equations of dynamical systems has been given for example by Marchuk [Marchuk95] [Marchuk96]. Specific examples of the adjoint model can be found in many different fields. Cacuci uses the *adjoint sensitivity formalism* to evaluate the partial derivatives of certain system responses with respect to thousands of input parameters [Cacuci81a] [Cacuci81b]. Talagrand gave an example of how this approach can be used for sensitivity calculation in meteorological applications [Talagrand91b]. Ustinov performed a sensitivity analysis based on the adjoint ERT applied to the case of atmospheric remote sensing in the thermal spectral region [Ustinov01]. Norton utilized the adjoint model for calculating the *Frechet* derivative of inverse scattering problems in neutron transport [Norton97] [Norton99].

In OT no group has implemented the adjoint transport equation in the calculation of the gradient within a MOBIIR scheme. However, Dorn used the time-dependent adjoint transport equation to determine the Frechet derivative of a residual that is proportional to the difference between predicted and measured data [Dorn98] [Dorn00]. The resulting nonlinear system of equations was solved by a nonlinear generalization of the ART, where the optical parameters are iteratively updated. He presented numerical results for scattering media with non-reentry boundary conditions.

A similar approach in OT was applied by Arridge *et al* to the diffusion equation [Arridge98]. Arridge derived the gradient $\nabla_\mu \Phi$ from the solution of the diffusion equation for a given source and from the solution of the adjoint diffusion equation for the boundary residual. The boundary residual is a function of the difference between the measured and predicted data. Since numerically solving the adjoint diffusion equation requires approximately the same amount of time as solving the diffusion equation itself, Arridge obtained the gradient in a time comparable to one forward calculation.

Several authors [Griewank89] [Sirkes97] [Kaminski99] have pointed out that the appropriate discretization scheme for the adjoint equation is in general different from the appropriate discretization scheme necessary for the forward equation. Therefore it is *a priori* not clear whether the gradient obtained with a discretized version of the adjoint equation truly equals the gradient of the discretized version of the forward equation. Therefore Shah [Shah91] and Talagrand *et al* [Talagrand87] have argued it is favorable to derive the adjoint model from the discretized form of the forward equation. This is the second way one can use adjoint schemes for calculating the gradient. This approach has been mainly applied to weather forecast models [Courtier87] and to ocean circulation models [Thacker88], but has not been pursued in OT.

The third approach for calculating the gradient by means of the adjoint model does not require the formulation of either a continuous or a discretized adjoint equation of the forward model. It is often referred to as computational differentiation in the adjoint or reverse mode, reverse differentiation, or adjoint differentiation [Rall81] [Rall91]. Here, the numerical code of the forward model, which is a sequence of arithmetic operations, is directly differentiated to compute the gradient. The procedure to find the derivatives of arbitrary algebraic functions, such as the gradient of an objective function, was first introduced by Wengert [Wengert64]. Over the last 15 years, Griewank [Griewank00] has generalized and refined the initial ideas in many publications on automatic differentiation [Rall91] [Beck94] [Coleman00], where the derivative is obtained by differentiating the forward code using an adjoint code compiler. Again, the main applications, so far, lie outside the field of OT as for example in geoscience [Talagrand91a] [Thacker91] [Kaminski99].

The key to this method is the decomposition of a given function, here the objective function containing the forward model, into a series of elementary differentiable functional steps. Then applying, systematically, the chain rule of differentiation to every single step of the forward code in the reverse direction, a numerical value for the gradient is obtained. The main advantage of this approach is that, at the level of the single steps in the forward model code, the gradient can be reconstructed according to simple rules [Kaminski99]. Thus the task can be handled without any explicit knowledge of the nature of the original problem.

Figure 6.1 depicts the relationships among the three different approaches to the gradient calculation using the adjoint model. All methods start from the forward model,

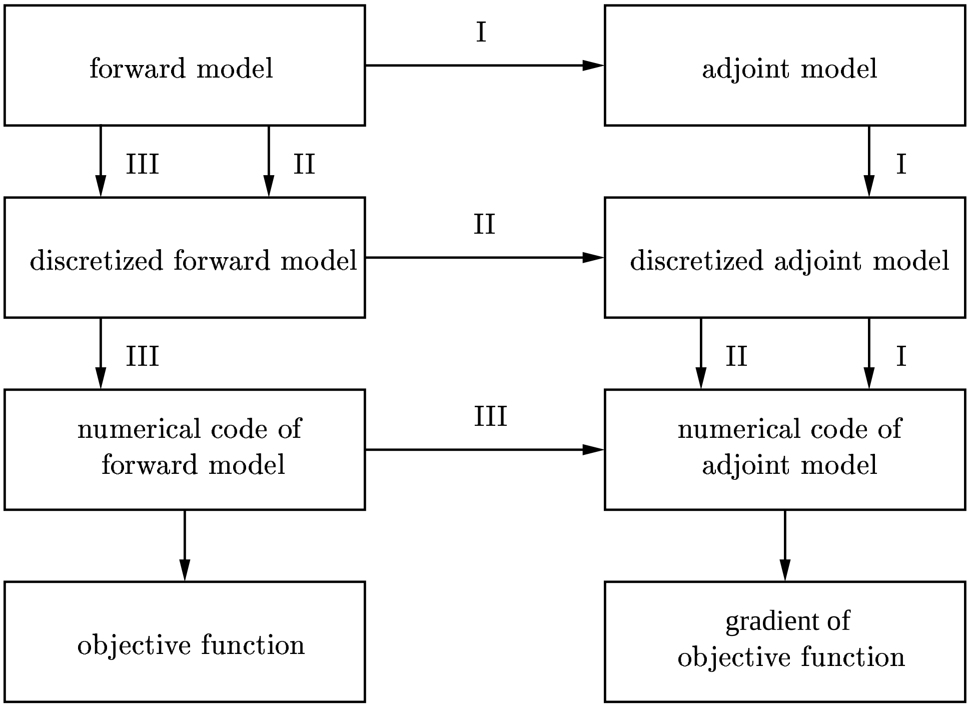
forward model
I
adjoint model
III
II
I
discretized forward model
II
discretized adjoint model
III
II
I
numerical code of forward model
III
numerical code of adjoint model
objective function
gradient of objective function

Figure 6.1: Three different ways of implementing the adjoint model to calculate the gradient of the objective function. The forward model can either be based on the diffusion equation or the ERT. Method III is represented by the adjoint differentiation technique.

which in OT is based on the diffusion equation or the ERT. In method I, one first derives the adjoint equation of the forward model, which is then discretized and numerically solved. Employing method II one first discretizes the equations used in the forward model, then derives the adjoint discretized model and solves it. Using method III, one first discretizes the forward model and solves it numerically. After that, the adjoint model is derived from the solution of the forward model. In the last case, the gradient is directly determined from the forward solution without any explicit knowledge of the adjoint equation.

In OT only the groups of Davis *et al* [Davies97], Roy *et al* [Roy99] and Hielscher *et al* [Hielscher99] have made use of the concept of adjoint differentiation. Davis and Roy have applied the adjoint differentiation technique to the time-independent diffusion equation and diffusion fluorescent problems in the frequency domain. Hielscher *et al* described an adjoint scheme for the time-dependent diffusion equation. In this work we have applied for the first time the adjoint differentiation technique to the ERT for deriving the gradient of the objective function.

6.3 Differentiation of Algorithms

The adjoint differentiation technique is part of a more general concept that is called *differentiation of algorithms* [Giering98] [Kaminski99]. This technique calculates the gradient of a function by applying the chain rule of differentiation. It can operate in the forward mode or in the reverse mode. We will explain this approach on the following example.

A function G maps the input variable $\boldsymbol{x}$ onto the output variable $\boldsymbol{y} = G(\boldsymbol{x})$ with :

$$\begin{aligned} G &: \mathbb{R}^{\mathrm{m}} \rightarrow \mathbb{R}^{\mathrm{n}} \\ \boldsymbol{x} &\mapsto \boldsymbol{y} = G(\boldsymbol{x}). \end{aligned} \tag{6.16}$$

Assuming the function G is for example a solution to a boundary-value problem and an analytical solution does not exist, the problem has to be solved numerically. A numerical implementation is represented by an algorithm, which maps the input variables onto the output variables. These algorithms usually consist of several sub-routines, which are executed in an iterative order. Therefore the function G is decomposed into Z differentiable sub-functions G^{z}:

$$\begin{aligned} G^{\mathrm{z}} &: \mathbb{R}^{\mathrm{m}_{\mathrm{z}-1}} \to \mathbb{R}^{\mathrm{m}_{\mathrm{z}}} \\ \boldsymbol{r}^{\mathrm{z}-1} &\mapsto \boldsymbol{r}^{\mathrm{z}} = G^{\mathrm{z}}(\boldsymbol{r}^{\mathrm{z}-1}), \end{aligned} \tag{6.17}$$

and we obtain

$$\boldsymbol{y} = G(\boldsymbol{x}) = (G^{\mathrm{Z}} \circ G^{\mathrm{Z}-1} \circ G^{\mathrm{Z}-2} \circ ... \circ G^2 \circ G^1)(\boldsymbol{x}). \tag{6.18}$$

An intermediate result $\boldsymbol{r}^{\mathrm{z}}$, also called a dependent variable, is

$$\boldsymbol{r}^{\mathrm{z}} = (G^{\mathrm{z}} \circ ... \circ G^2 \circ G^1)(\boldsymbol{x}). \tag{6.19}$$

Subsequently, the intermediate result $\boldsymbol{r}^{\mathrm{z}}$ becomes an input variable for the next operation $G^{\mathrm{z}+1}(\boldsymbol{r}^{\mathrm{z}})$. The final result for the forward problem is calculated by stepping forward through all intermediate steps. The intermediate dependent variables $\boldsymbol{r}^{\mathrm{z}}$ are a result of the

$$\boldsymbol{x} \xrightarrow{G^1} \boldsymbol{r}^1 \xrightarrow{G^2} \boldsymbol{r}^2 \;\ldots\; \boldsymbol{r}^{\mathrm{Z}-3} \xrightarrow{G^{\mathrm{Z}-2}} \boldsymbol{r}^{\mathrm{Z}-2} \xrightarrow{G^{\mathrm{Z}-1}} \boldsymbol{r}^{\mathrm{Z}-1} \xrightarrow{G^{\mathrm{Z}}} \boldsymbol{y}$$

Figure 6.2: Computational graph of the function evaluation by stepping through all sub-functions from left to right.

decomposition of the function. The decomposition can also be displayed by a computational graph (see Figure 6.2). It shows in which order the solution is evaluated by stepping through the graph.

The derivative of the function G is computed using the chain rule of differentiation for a composite function from the individual derivatives of the sub-functions G^{z}. These derivatives are in general Jacobian matrices $\boldsymbol{\mathcal{G}}$. The resulting product of Jacobian matrices can be evaluated in any order, since matrix multiplication is an associative operation. Operating in the forward mode, the intermediate derivatives are evaluated in the same order (see Figure 6.3) as the forward model computes the intermediate variables:

$$\boldsymbol{\mathcal{G}} = \frac{\partial G(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial G^{\mathrm{Z}}(\boldsymbol{r}^{\mathrm{Z}-1})}{\partial \boldsymbol{r}^{\mathrm{Z}-1}} \circ \ldots \circ \frac{\partial G^{3}(\boldsymbol{r}^{2})}{\partial \boldsymbol{r}^{2}} \circ \frac{\partial G^{2}(\boldsymbol{r}^{1})}{\partial \boldsymbol{r}^{1}} \circ \frac{\partial G^{1}(\boldsymbol{x})}{\partial \boldsymbol{x}}. \tag{6.20}$$

At the z-th step of the forward mode we get the intermediate derivative:

$$\frac{\partial (G^{\mathrm{z}} \circ \ldots \circ G^{1})(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial G^{\mathrm{z}}(\boldsymbol{r}^{\mathrm{z}-1})}{\partial \boldsymbol{r}^{\mathrm{z}-1}} \frac{\partial (G^{\mathrm{z}-1} \circ \ldots \circ G^{1})(\boldsymbol{x})}{\partial \boldsymbol{x}}. \tag{6.21}$$

$$\begin{tikzcd}
\frac{\partial G^1}{\partial \boldsymbol{x}} \arrow[r, "\frac{\partial G^2}{\partial \boldsymbol{r}^1}"] & \frac{\partial G^2}{\partial \boldsymbol{x}} \arrow[r, "\frac{\partial G^3}{\partial \boldsymbol{r}^2}"] & \frac{\partial G^3}{\partial \boldsymbol{x}} & \cdots & \frac{\partial G^{\mathrm{Z}-3}}{\partial \boldsymbol{x}} \arrow[r, "\frac{\partial G^{\mathrm{Z}-2}}{\partial \boldsymbol{r}^{\mathrm{Z}-3}}"] & \frac{\partial G^{\mathrm{Z}-2}}{\partial \boldsymbol{x}} \arrow[r, "\frac{\partial G^{\mathrm{Z}-1}}{\partial \boldsymbol{r}^{\mathrm{Z}-2}}"] & \frac{\partial G^{\mathrm{Z}-1}}{\partial \boldsymbol{x}} \arrow[r, "\frac{\partial G^{\mathrm{Z}}}{\partial \boldsymbol{r}^{\mathrm{Z}-1}}"] & \frac{\partial G}{\partial \boldsymbol{x}} = \frac{\partial G^{\mathrm{Z}}}{\partial \boldsymbol{x}}
\end{tikzcd}$$

Figure 6.3: Computational graph of the forward mode of differentiation of algorithms. The derivative $\frac{\partial G}{\partial \boldsymbol{x}}$ is evaluated by stepping from left to right.

In the reverse mode the operations are computed from right to left (see Figure 6.4). For real-valued elements of the matrix $\boldsymbol{\mathcal{G}}$ the adjoint matrix $\boldsymbol{\mathcal{G}}^{*}$ is just the transposed matrix $\boldsymbol{\mathcal{G}}^{T}$:

$$\boldsymbol{\mathcal{G}}^{T} = \frac{\partial G(\boldsymbol{x})}{\partial \boldsymbol{x}}^{T} = \frac{\partial G^{1}(\boldsymbol{x})}{\partial \boldsymbol{x}}^{T} \circ \frac{\partial G^{2}(\boldsymbol{r}^{1})}{\partial \boldsymbol{r}^{1}}^{T} \circ \frac{\partial G^{3}(\boldsymbol{r}^{2})}{\partial \boldsymbol{r}^{2}}^{T} \circ \ldots \circ \frac{\partial G^{\mathrm{Z}}(\boldsymbol{r}^{\mathrm{Z}-1})}{\partial \boldsymbol{r}^{\mathrm{Z}-1}}^{T}. \tag{6.22}$$

Corresponding to the z-th step, the composition of the intermediate transposed Jacobian matrix is evaluated by multiplying the intermediate transpose Jacobian matrix from the $(\mathrm{z}+1)$-th step $\frac{\partial(G^{\mathrm{Z}}\circ\ldots\circ G^{\mathrm{z}+1})(\boldsymbol{r}^{\mathrm{z}})}{\partial \boldsymbol{r}^{\mathrm{z}}}^T$ by the transpose $\frac{\partial G^{\mathrm{z}}}{\partial \boldsymbol{r}^{\mathrm{z}-1}}^T$ and we obtain:

$$\frac{\partial(G^{\mathrm{Z}}\circ\ldots\circ G^{\mathrm{z}})(\boldsymbol{r}^{\mathrm{z}-1})}{\partial \boldsymbol{r}^{\mathrm{z}-1}}^T = \frac{\partial G^{\mathrm{z}}(\boldsymbol{r}^{\mathrm{z}-1})}{\partial \boldsymbol{r}^{\mathrm{z}-1}}^T \frac{\partial(G^{\mathrm{Z}}\circ\ldots\circ G^{\mathrm{z}+1})(\boldsymbol{r}^{\mathrm{z}})}{\partial \boldsymbol{r}^{\mathrm{z}}}^T . \tag{6.23}$$

$$\begin{tikzcd}
\frac{\partial G}{\partial \boldsymbol{x}}^T & \frac{\partial G}{\partial \boldsymbol{r}^1}^T \arrow[l, "\frac{\partial G^1}{\partial \boldsymbol{x}}^T"'] & \frac{\partial G}{\partial \boldsymbol{r}^2}^T \arrow[l, "\frac{\partial G^2}{\partial \boldsymbol{r}^1}^T"'] & \cdots & \frac{\partial G}{\partial \boldsymbol{r}^{\mathrm{Z}-3}}^T & \frac{\partial G}{\partial \boldsymbol{r}^{\mathrm{Z}-2}}^T \arrow[l, "\frac{\partial G^{\mathrm{Z}-2}}{\partial \boldsymbol{r}^{\mathrm{Z}-3}}^T"'] & \frac{\partial G}{\partial \boldsymbol{r}^{\mathrm{Z}-1}}^T \arrow[l, "\frac{\partial G^{\mathrm{Z}-1}}{\partial \boldsymbol{r}^{\mathrm{Z}-2}}^T"'] & 1 \arrow[l, "\frac{\partial G^{\mathrm{Z}}}{\partial \boldsymbol{r}^{\mathrm{Z}-1}}^T"']
\end{tikzcd}$$

Figure 6.4: Computational graph of the reverse or adjoint mode of differentiation of algorithms. The derivative $\frac{\partial G}{\partial \boldsymbol{x}}^T$ is evaluated by stepping backwards from right to left.

Depending on the size of the Jacobian matrices one mode is more computationally efficient than the other. If the number of output variables exceeds the number of input variables the forward mode needs less computational operations than the reverse mode. If the number of input variables exceeds the number of output variables then the reverse mode of differentiation is computationally more efficient. An example of differentiation of algorithms in the forward and reverse mode is given in Appendix B.

We apply the differentiation of algorithms to the objective function $\Phi(\boldsymbol{\mu})$ in OT, which will result in the gradient of the objective function. Here, the number of input variables, given by the vector $\boldsymbol{\mu}$ with $\mathrm{N} = 2 \times \mathrm{I} \times \mathrm{J}$ entities, exceeds the number of output variables, which is just a single value $\tilde{\varphi}$ of the objective function $\Phi(\boldsymbol{\mu})$ (see also Definitions 5.1 and 5.2). The reverse mode of differentiation is therefore more suitable for deriving the gradient. This will be subject of the next section.

6.4 Adjoint Differentiation of the Objective Function

After the general overview of the adjoint model and its numerical implementation just presented, we are ready to apply the adjoint differentiation technique to the objective function in OT. First, we decompose the objective function into a series of elementary differentiable functional steps. These functional steps are given by the forward model based on the ERT. Then by systematically applying the chain rule of differentiation to every single step of the forward code in the reverse direction, the gradient of the objective function is obtained.

6.4.1 Decomposition of the Forward Model

The objective function Φ is a composite function of the optical parameters $\boldsymbol{\mu}$ with $\Phi(\boldsymbol{\mu}) = \tilde{\Phi}(\boldsymbol{p}(\boldsymbol{\mu}))$. It can be decomposed into sub-functions according to Equation 6.18 in the following way:

$$\begin{aligned} \Phi(\boldsymbol{\mu}) &= \tilde{\Phi}\left(F^{\mathrm{Z}}\left(F^{\mathrm{Z}-1}\left(F^{\mathrm{Z}-2}\left(\ldots\left(F^{2}\left(F^{1}\left(\boldsymbol{\mu}\right),\boldsymbol{\mu}\right)\right)\ldots\right),\boldsymbol{\mu}\right),\boldsymbol{\mu}\right)\right) \\ &:= \left(\tilde{\Phi}\circ F^{\mathrm{Z}}(\boldsymbol{\mu})\circ F^{\mathrm{Z}-1}(\boldsymbol{\mu})\circ F^{\mathrm{Z}-2}(\boldsymbol{\mu})\circ\ldots\circ F^{2}(\boldsymbol{\mu})\circ F^{1}\right)(\boldsymbol{\mu})\,. \end{aligned} \tag{6.24}$$

Each sub-function is an explicit function of $\boldsymbol{\mu}$ again, which makes it different from Equation 6.18. The sub-functions F^{z} are given by the successive iteration steps in the SOR method, which is used to solve the forward model. The SOR method is an iterative approach and the z-th iteration yields the intermediate result $\psi^{\mathrm{z}}_{\mathrm{kij}}$. The radiance vector $\boldsymbol{\psi}$ consists of $\mathrm{M} = \mathrm{I}\times\mathrm{J}\times\mathrm{K}$ elements with $\mathrm{i}\in[1,\mathrm{I}]$, $\mathrm{j}\in[1,\mathrm{J}]$, and $\mathrm{k}\in[1,\mathrm{K}]$. The detector readings $\boldsymbol{p}$ are the angular-dependent radiances $\psi^{\mathrm{Z}}_{\mathrm{kij}}$ at the last iteration step Z at detector positions (i,j) on the boundary. The computational graph of the transport forward model is depicted in Figure 6.5.

A value $\tilde{\varphi}$ of the objective function is evaluated by stepping through the computa-

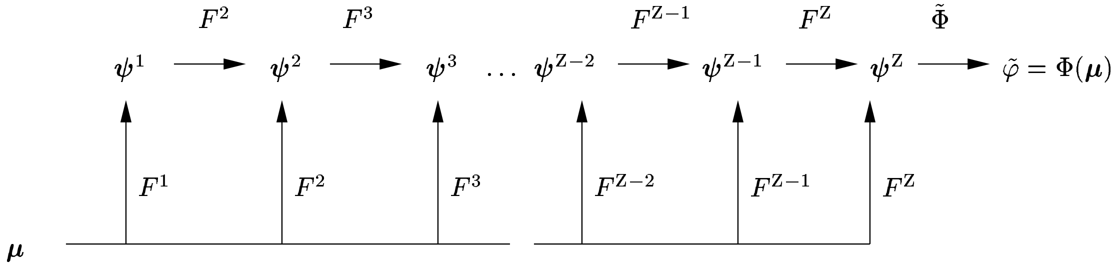

Figure 6.5: Computational graph of the transport forward model. The objective function Φ is calculated by stepping through all sub-functions from left to right. The sub-functions are given by the SOR method for solving the discretized ERT.

tional graph. Starting with the optical parameters $\boldsymbol{\mu}$ as the input variables, the sub-function F^1 produces the intermediate result and output variable $\boldsymbol{\psi}^1$.

$$F^1 : \mathbb{R}^{\mathrm{N}} \to \mathbb{R}^{\mathrm{M}}$$

$$\boldsymbol{\mu} \mapsto \boldsymbol{\psi}^1, \tag{6.25}$$

and we have for example for the ordinates with $\xi_{\mathrm{k}} > 0, \eta_{\mathrm{k}} > 0$

$$\psi^1_{\mathrm{kij}} = \rho \frac{S_{\mathrm{kij}} + \frac{\xi_{\mathrm{k}}}{\triangle x}\psi^1_{\mathrm{ki-1j}} + \frac{\eta_{\mathrm{k}}}{\triangle y}\psi^1_{\mathrm{kij-1}}}{\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_a]_{\mathrm{ij}} + [\mu_s]_{\mathrm{ij}}}. \tag{6.26}$$

The sub-functions F^{z} map the intermediate variables $\boldsymbol{\psi}^{\mathrm{z}-1}$ and input variables $\boldsymbol{\mu}$ onto the intermediate result $\boldsymbol{\psi}^{\mathrm{z}} = F^{\mathrm{z}}(\boldsymbol{\psi}^{\mathrm{z}-1}, \boldsymbol{\mu})$ for all iteration steps of the transport forward model:

$$F^{\mathrm{z}} : \mathbb{R}^{\mathrm{M}} \times \mathbb{R}^{\mathrm{N}} \to \mathbb{R}^{\mathrm{M}}$$

$$\begin{pmatrix} \boldsymbol{\psi}^{\mathrm{z}-1} \\ \boldsymbol{\mu} \end{pmatrix} \mapsto \boldsymbol{\psi}^{\mathrm{z}}. \tag{6.27}$$

This mapping is shown explicitly for the ordinates with $\xi_{\mathrm{k}} > 0, \eta_{\mathrm{k}} > 0$:

$$\psi^{\mathrm{z}}_{\mathrm{kij}} = (1-\rho)\psi^{\mathrm{z}-1}_{\mathrm{kij}} + \rho \frac{S_{\mathrm{kij}} + [\mu_s]_{\mathrm{ij}} \sum_{\mathrm{k}'} a_{\mathrm{k}'} \tilde{p}_{\mathrm{kk}'} \psi^{\mathrm{z}-1}_{\mathrm{k}'\mathrm{ij}} + \frac{\xi_{\mathrm{k}}}{\triangle x}\psi^{\mathrm{z}}_{\mathrm{ki-1j}} + \frac{\eta_{\mathrm{k}}}{\triangle y}\psi^{\mathrm{z}}_{\mathrm{kij-1}}}{\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_a]_{\mathrm{ij}} + [\mu_s]_{\mathrm{ij}}}. \tag{6.28}$$

The last step F^{Z} calculates the predictions $\boldsymbol{p}$, which become the input to the final step of $\tilde{\Phi}$, which is the calculation of the scalar $\tilde{\varphi}$. Equations 6.26 and 6.28 are the smallest computational units in the transport forward model.

6.4.2 Adjoint Differentiation

To obtain the gradient of the objective function we start differentiating 6.24 with respect to the optical parameters $\boldsymbol{\mu}$. We derived the following expression for the gradient (see Appendix C):

$$\nabla_{\boldsymbol{\mu}} \Phi^T = \frac{\partial \boldsymbol{\psi}^{1^T}}{\partial \boldsymbol{\mu}} \left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^1} \right)^T + \frac{\partial \boldsymbol{\psi}^{2^T}}{\partial \boldsymbol{\mu}} \left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^2} \right)^T + \ldots + \frac{\partial \boldsymbol{\psi}^{\mathrm{Z}^T}}{\partial \boldsymbol{\mu}} \left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{Z}}} \right)^T . \tag{6.29}$$

The terms $\frac{\partial \boldsymbol{\psi}^{\mathrm{z}}}{\partial \boldsymbol{\mu}}$ can be calculated from Equation 6.28 of the forward model. For the derivatives with respect to $[\mu_s]_{\mathrm{ij}}$ and $[\mu_a]_{\mathrm{ij}}$ we obtain

$$\left[\frac{\partial \psi^{\mathrm{z}}}{\partial \mu_s} \right]_{\mathrm{kij}} = \rho \frac{\sum_{\mathrm{k}'} a_{\mathrm{k}'} \tilde{p}_{\mathrm{kk}'} \psi^{\mathrm{z}-1}_{\mathrm{k}'\mathrm{ij}}}{\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_a]_{\mathrm{ij}} + [\mu_s]_{\mathrm{ij}}} - \rho \frac{S_{\mathrm{kij}} + [\mu_s]_{\mathrm{ij}} \sum_{\mathrm{k}'} a_{\mathrm{k}'} \tilde{p}_{\mathrm{kk}'} \psi^{\mathrm{z}-1}_{\mathrm{k}'\mathrm{ij}} + \frac{\xi_{\mathrm{k}}}{\triangle x} \psi^{\mathrm{z}}_{\mathrm{ki-1j}} + \frac{\eta_{\mathrm{k}}}{\triangle y} \psi^{\mathrm{z}}_{\mathrm{kij-1}}}{\left(\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_a]_{\mathrm{ij}} + [\mu_s]_{\mathrm{ij}} \right)^2} \tag{6.30}$$

and

$$\left[\frac{\partial \psi^{\mathrm{z}}}{\partial \mu_a} \right]_{\mathrm{kij}} = -\rho \frac{S_{\mathrm{kij}} + [\mu_s]_{\mathrm{ij}} \sum_{\mathrm{k}'} a_{\mathrm{k}'} \tilde{p}_{\mathrm{kk}'} \psi^{\mathrm{z}-1}_{\mathrm{k}'\mathrm{ij}} + \frac{\xi_{\mathrm{k}}}{\triangle x} \psi^{\mathrm{z}}_{\mathrm{ki-1j}} + \frac{\eta_{\mathrm{k}}}{\triangle y} \psi^{\mathrm{z}}_{\mathrm{kij-1}}}{\left(\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_a]_{\mathrm{ij}} + [\mu_s]_{\mathrm{ij}} \right)^2} . \tag{6.31}$$

At this point we have not yet used the adjoint differentiation technique, since we have not stepped backward through the forward code. This procedure comes into play in the calculation of the terms $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{z}}} \right)^T$ in Equation 6.29 by using Equation 6.23 of the adjoint mode of differentiation. This can be best understood while looking at the computational graph of the reverse mode of differentiation in Figure 6.6. Starting with the last step of the forward code, which is the calculation of the objective function (see Equation 5.1) given the predictions $\boldsymbol{p} = \boldsymbol{\psi}^{\mathrm{Z}}$, we differentiate Φ with respect to $\boldsymbol{\psi}^{\mathrm{Z}}$. The result is the difference

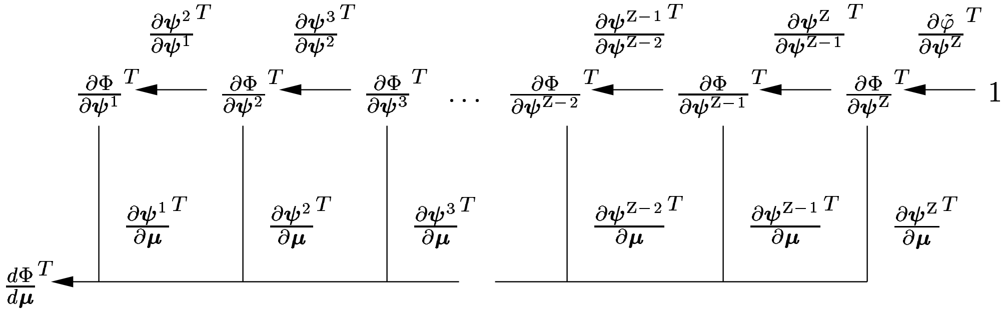
$\frac{\partial \boldsymbol{\psi}^2}{\partial \boldsymbol{\psi}^1}^T$
$\frac{\partial \boldsymbol{\psi}^3}{\partial \boldsymbol{\psi}^2}^T$
$\frac{\partial \boldsymbol{\psi}^{Z-1}}{\partial \boldsymbol{\psi}^{Z-2}}^T$
$\frac{\partial \boldsymbol{\psi}^{Z}}{\partial \boldsymbol{\psi}^{Z-1}}^T$
$\frac{\partial \tilde{\varphi}}{\partial \boldsymbol{\psi}^{Z}}^T$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^1}^T$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^2}^T$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^3}^T$
$\cdots$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{Z-2}}^T$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{Z-1}}^T$
$\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{Z}}^T$
1
$\frac{\partial \boldsymbol{\psi}^1}{\partial \boldsymbol{\mu}}^T$
$\frac{\partial \boldsymbol{\psi}^2}{\partial \boldsymbol{\mu}}^T$
$\frac{\partial \boldsymbol{\psi}^3}{\partial \boldsymbol{\mu}}^T$
$\frac{\partial \boldsymbol{\psi}^{Z-2}}{\partial \boldsymbol{\mu}}^T$
$\frac{\partial \boldsymbol{\psi}^{Z-1}}{\partial \boldsymbol{\mu}}^T$
$\frac{\partial \boldsymbol{\psi}^{Z}}{\partial \boldsymbol{\mu}}^T$
$\frac{d\Phi}{d\boldsymbol{\mu}}^T$

Figure 6.6: Computational graph of the adjoint differentiation technique applied to the transport forward model. The derivative $\frac{d\Phi}{d\boldsymbol{\mu}}^T$ is calculated by stepping backwards through the computational graph of the forward model (see Figure 6.5).

between the measured and predicted data for all D source-detector pairs:

$$\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{Z}}}\right)^T = \frac{1}{\kappa^2}(\boldsymbol{\psi}^{\mathrm{Z}} - \boldsymbol{m})^T. \tag{6.32}$$

Equation 6.32 is the input parameter $\nabla_{\boldsymbol{p}}\Phi$ to the adjoint model, which will eventually provide the output parameter $\nabla_{\boldsymbol{\mu}}\Phi$ (see Equation 6.15). More specifically, continuing to step backward through the forward code we calculate $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{Z}-1}}\right)^T$, which is given by

$$\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{Z}-1}}\right)^T = \left(\frac{\partial \boldsymbol{\psi}^{Z}}{\partial \boldsymbol{\psi}^{\mathrm{Z}-1}}\right)^T \left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{Z}}}\right)^T. \tag{6.33}$$

The remaining derivatives $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{z}}}\right)^T$ of all intermediate steps in Equation 6.29 are computed recursively using the previously calculated derivatives $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{z}+1}}\right)^T$. This step, in which $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{z}}}\right)^T$ is calculated from $\left(\frac{\partial \Phi}{\partial \boldsymbol{\psi}^{\mathrm{z}+1}}\right)^T$, constitutes the adjoint differentiation step. The matrix $\left(\frac{\partial \boldsymbol{\psi}^{\mathrm{z}+1}}{\partial \boldsymbol{\psi}^{\mathrm{z}}}\right)^T$ is obtained by differentiating the (z + 1)-th SOR-iteration step, given in Equation 6.28, with respect to $\boldsymbol{\psi}^{\mathrm{z}}$. We get, for example in the case of the ordinates with $\xi_{\mathrm{k}} > 0$ and $\eta_{\mathrm{k}} > 0$:

$$\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{kij}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}} = (1-\rho)\delta_{\mathrm{kij}} + \rho\frac{[\mu_s]_{\mathrm{ij}}a_{\mathrm{k'}}\tilde{p}_{\mathrm{kk'}}\delta_{\mathrm{ij}} + \frac{\xi_{\mathrm{k}}}{\triangle x}\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{ki-1j}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}} + \frac{\eta_{\mathrm{k}}}{\triangle y}\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{kij-1}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}}}{\frac{\xi_{\mathrm{k}}}{\triangle x} + \frac{\eta_{\mathrm{k}}}{\triangle y} + [\mu_s]_{\mathrm{ij}} + [\mu_a]_{\mathrm{ij}}} \tag{6.34}$$

with

$$\delta_{\mathrm{kij}} = \delta_{\mathrm{k}}\delta_{\mathrm{i}}\delta_{\mathrm{j}} \text{ with } \delta_a = \begin{cases} 1 & \text{if } a' = a \\ 0 & \text{if } a' \neq a. \end{cases}$$

The derivatives $\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{ki-1j}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}}$ and $\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{kij-1}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}}$ in Equation 6.34 are obtained by differentiating Equation 6.28 again. However, we made the approximations

$$\frac{\xi_{\mathrm{k}}}{\triangle x}\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{ki-1j}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}} := \frac{\xi_{\mathrm{k}}}{\triangle x}\delta_{\mathrm{ki-1j}} \tag{6.35}$$

and

$$\frac{\eta_{\mathrm{k}}}{\triangle y}\frac{\partial\psi^{\mathrm{z+1}}_{\mathrm{kij-1}}}{\partial\psi^{\mathrm{z}}_{\mathrm{k'i'j'}}} := \frac{\eta_{\mathrm{k}}}{\triangle y}\delta_{\mathrm{kij-1}} \tag{6.36}$$

for the relevant terms on the right-hand side of Equation 6.34, because $\psi^{\mathrm{z+1}}_{\mathrm{ki-1j}}$ and $\psi^{\mathrm{z+1}}_{\mathrm{kij-1}}$ are slowly varying functions of $\psi^{\mathrm{z}}_{\mathrm{kij}}$.

As can be seen, the gradient of the objective function is calculated stepping backwards through all previously calculated iteration steps of the forward model without solving an entirely new numerical problem of the adjoint ERT. Furthermore, the particular underlying physical system does not have to be known, because the derivative is computed directly from the code of the forward model (Equations 6.26 and 6.28). A disadvantage of the reverse mode of differentiation is that all intermediate results $\psi^{\mathrm{z}}_{\mathrm{kij}}$ of the forward model had to be stored for subsequent use in the reverse mode [3].

[3] Example: An I × J = 60 × 60 grid with K=16 ordinates, and Z=300 iteration steps of the SOR method, requires approximately 132 MByte storage space for all elements $\psi^{\mathrm{z}}_{\mathrm{kij}}$ with 8 Byte each element.

6.5 Scaling Factor

The gradient $\nabla_\mu \Phi$ is used within an optimization technique as discussed in Chapter 5 for calculating the search direction $\boldsymbol{u}_k$. The minimum is found by employing a line search technique. The line search technique determines a step length α_k for updating the optical parameters along the search direction (see Equation 5.4). However, the length of the gradient ($\|\nabla_\mu \Phi\|$) can vary between 10^{-6} and 10^3 depending on the particular optical parameters of the medium and the initial guess $\boldsymbol{\mu}_0$. Not having a unit length, the gradient will severely influence the determination of the step length within the line search. This can lead to a premature convergence of the optimization technique, because the line search fails to find the proper step length. Consequently, the gradient $\nabla_\mu \Phi$ has to be scaled in such a way that the line search proceeds independently of the gradient length. We have found that we obtained significant better updates of the optical parameters when using a scaled gradient vector $\nabla_\mu \Phi^{scaled}$. Both components, $\nabla_{\mu_s} \Phi$ and $\nabla_{\mu_a} \Phi$, of the gradient vector were scaled independently by using the scaling factors χ_s and χ_a with

$$\nabla_{\mu_s} \Phi^{scaled} = \chi_s \nabla_{\mu_s} \Phi \tag{6.37}$$

$$\nabla_{\mu_a} \Phi^{scaled} = \chi_a \nabla_{\mu_a} \Phi. \tag{6.38}$$

We have empirically chosen the scaling factor χ_s after the first iteration k=1 such that the largest element of the scaled gradient vector $\nabla_{\mu_s} \Phi^{scaled}$ equals 5% of the largest element of the vector $\boldsymbol{\mu}_{s_0}$:

$$\chi_s = 0.05 \frac{\max(\mu_{\mathrm{s}0_\mathrm{i}})}{\max(|\nabla_{\mu_\mathrm{s}} \Phi_\mathrm{i}|)}. \tag{6.39}$$

The same holds for χ_a. The factors χ_s and χ_a were maintained constant throughout the optimization process after the first iteration.

6.6 Example of a Derivative Calculation Based on Experimental Data

The gradient $\nabla_{\boldsymbol{\mu}}\Phi$ can be displayed in a two-dimensional image. It tells us how the location and strength of the spatially distributed optical parameters of a medium, represented by the vector $\boldsymbol{\mu}$, deviate from a given initial guess $\boldsymbol{\mu}_0$ when measurements $\boldsymbol{m}$ were taken on the boundary of the medium and the prediction $\boldsymbol{p}(\boldsymbol{\mu}_0)$ were calculated. As an example we demonstrate how the derivative information was obtained by performing measurements on the boundary of a scattering phantom.

A scattering phantom was designed as depicted in Figure 6.7. It had dimensions of 3 cm × 3 cm × 14 cm and contained a cylindrical hole of a diameter of 0.5 cm. The hole was filled with a scattering fluid (INTRALIPID®) with a $\mu_{\mathrm{s}}' = 23.2 \pm 5\ \mathrm{cm}^{-1}$ and a $\mu_{\mathrm{a}} = 0.00675 \pm 0.003\ \mathrm{cm}^{-1}$. These optical parameters were determined by a formula for INTRALIPID® given by Flock *et al* [Flock89a] [Flock89b]. More details on the phantom material and the experimental set-up were given in Chapter 4.

Three sources were placed on each side of the phantom. The movable detector was located on the boundary of the side opposite to the source. We recorded 28 detector points for each source point yielding a measurement vector $\boldsymbol{m}$ with a total of $\mathrm{D} = 12 \times 28$ source-detector pairs. The scattering phantom with its source-detector configuration is given in Figure 6.7.

The detector predictions $\boldsymbol{p}$ were calculated by the transport forward model assuming a homogeneous distribution of $\mu_{\mathrm{s}} = 50\ \mathrm{cm}^{-1}$, $\mu_{\mathrm{a}} = 0.45\ \mathrm{cm}^{-1}$, and $g = 0.86$. The calculations were performed on a 61 × 61 grid with 16 discrete ordinates. In Figure 6.8, we present an example of the predictions and actual experimental detector readings for source A, as given in Figure 6.7. Due to the highly scattering cylindrical perturbation, more

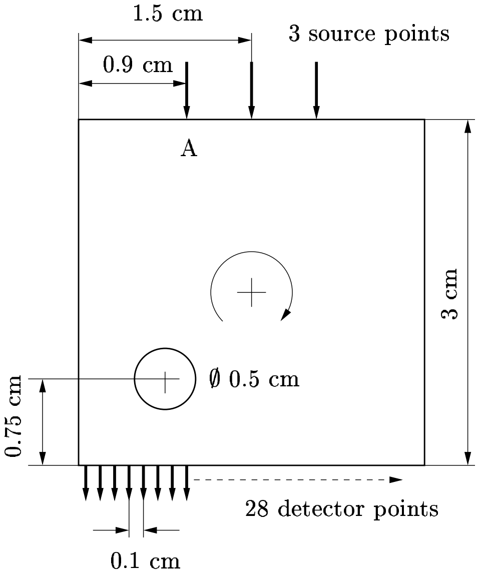

Figure 6.7: Schematic and source-detector configuration of the phantom that contained a single scattering heterogeneity. The phantom was illuminated from all four sides. The measurements were taken on the sides opposite the sources.

photons get scattered. Thus, fewer photons reach the detectors at positions $x = 0.4$ cm through $x = 1.3$ cm (see Figure 6.8).

A value $\tilde{\varphi}$ of the objective function Φ is determined using Definition 5.1 given the measurement vector $\boldsymbol{m}$ and the prediction vector $\boldsymbol{p}$. The gradient $\nabla_{\mu_s}\Phi$ of the objective function Φ with respect to the scattering coefficients μ_s is computed using the adjoint differentiation technique as explained in Subsection 6.4.2.

The scaled gradient $\nabla_{\boldsymbol{\mu}_s}\Phi$ is shown in Figure 6.9. It depicts the change of the objective function to changes in the optical parameters of the initial guess μ_{s_0}. The distance between adjacent isolines is 0.005. Values $\nabla_{\boldsymbol{\mu}_s}\Phi_{\mathrm{i}}$ of the gradient vector are in the range $[-0.05 — 0.03]$. The homogeneous medium is depicted by values $\nabla_{\boldsymbol{\mu}_s}\Phi_{\mathrm{i}} \approx 0$, where the assumed scattering coefficients $\boldsymbol{\mu}_{s_0}$ of the forward model match the scattering coefficients $\boldsymbol{\mu}_s$ of the phantom. The scattering heterogeneity is visible in the lower left corner of Figure 6.9

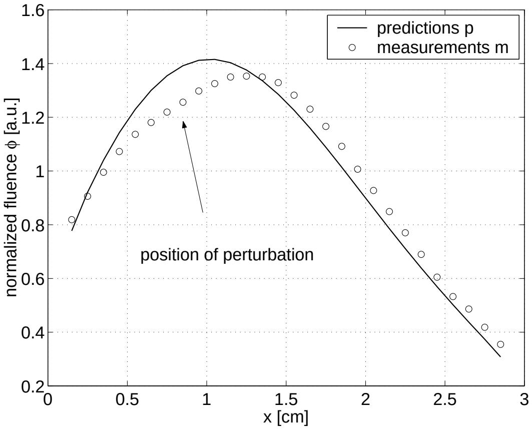

Figure 6.8: Comparison of predictions and measurements for source A. The predictions were calculated by assuming a homogeneous medium, whereas measurements were performed on the phantom containing a scattering heterogeneity (see Figure 6.7).

with values $\nabla_{\boldsymbol{\mu}_s}\Phi_{\mathrm{i}}$ in the range $[-0.05 - -0.02]$.

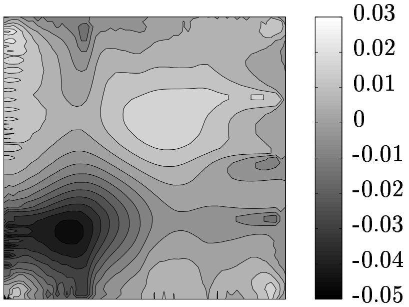

Figure 6.9: Scaled gradient $\nabla_{\boldsymbol{\mu}_s}\Phi$ of the objective function with respect to the scattering coefficient. It depicts alterations of the assumed homogeneous distribution of the scattering coefficient μ_{s_0} to the scattering coefficients μ_s of the original medium.

Once the gradient $\nabla_{\boldsymbol{\mu}}\Phi$ is calculated, it is subsequently used for determining the search direction $\boldsymbol{u}_{\mathrm{k}}$ within the numerical optimization technique (Equations 5.16, 5.30, and 5.32). It is obvious that negative components of the gradient vector ($\boldsymbol{u}_{\mathrm{k}} \sim -\nabla_{\mu}\Phi$, see Equations 5.16, 5.30, and 5.32) lead to a positive update of the optical parameters by using the update formula given by Equation 5.4.