Regula falsi based automatic regularization method for PDE constrained optimization

Regula falsi based automatic regularization method for PDE constrained optimization

Nick Schenkels[1], Wim Vanroose[2]

Departement of Mathematics and Computer Science,
University of Antwerp,
Antwerp, Belgium

Abstract

Many inverse problems can be described by a PDE model with unknown parameters that need to be calibrated based on measurements related to its solution. This can be seen as a constrained minimization problem where one wishes to minimize the mismatch between the observed data and the model predictions, including an extra regularization term, and use the PDE as a constraint. Often, a suitable regularization parameter is determined by solving the problem for a whole range of parameters – e.g. using the L-curve – which is computationally very expensive. In this paper we derive two methods that simultaneously solve the inverse problem and determine a suitable value for the regularization parameter. The first one is a direct generalization of the Generalized Arnoldi Tikhonov method for linear inverse problems. The second method is a novel method based on similar ideas, but with a number of advantages for nonlinear problems.

Keywords: PDE constrained optimization, regularization, Morozov's discrepancy principle, Newton-Krylov, inverse scattering.

1. Introduction

The dynamics of many complex applications are described by a PDE model $F(u, k) = 0$, with solution or state variables u and parameters or control variables k. Examples include all forms of wave scattering problems [1, 2], various financial models [3, 4], etc. The forward problem, i.e. solving the PDE for u given the parameters k, is often well understood and is in many cases solved fast and accurately by a numerical method. The inverse problem, i.e. finding the parameters k such that the solution u matches a set of observations $\widetilde{u}$ as best as possible, is, however, much more complicated because these problems are typically ill-posed.

[1]Corresponding author: nick.schenkels@uantwerpen.be
[2]wim.vanroose@uantwerpen.be

Let $H(k) \in \mathbb{C}^{m\times n}$, $u \in \mathbb{C}^n$, $f(k) \in \mathbb{C}^m$ and $k \in \mathbb{C}^l$ be such that

$$H(k)u = f(k), \tag{1}$$

is the discretized version of the PDE with the appropriate initial and boundary conditions. If $\widetilde{u} \in \mathbb{C}^p$ are the observations of the solution of the PDE and assuming that u is an implicit function of k, we consider the following constrained optimization problem:

$$\begin{cases} \min\limits_{k\in\mathbb{C}^l} \mathcal{J}(k) = \min\limits_{k\in\mathbb{C}^l} \underbrace{\|Lu-\widetilde{u}\|^2}_{\mathcal{D}(k):=} + \alpha \underbrace{\|k-k_0\|^2}_{\mathcal{R}(k):=} \\ \quad H(k)u = f(k). \end{cases} \tag{2}$$

Here, $\|\cdot\|$ denotes the standard Euclidian norm, $L \in \mathbb{R}^{p\times n}$ is a linear operator that maps the full solution u of the PDE to the observed output $\widetilde{u}$, $\mathcal{D}(k)$ is a discrepancy or residual term measuring the mismatch between the model predictions u and the observations $\widetilde{u}$ and $\mathcal{R}(k)$ is the regularization term added in order to place certain constraints on the parameters, incorporate prior knowledge, suppress numerical errors or guarantee that the problem is well-posed.

There are now two difficulties, the first of which is calculating the gradient of $\mathcal{J}(k)$. This is necessary because many nonlinear optimization algorithms use some form of gradient information, e.g. Newton's method, steepest descent, nonlinear CG, etc [5]. However, approximating $\nabla\mathcal{J}(k)$ using finite difference methods is inefficient when, for example, k is very high dimensional [6]. In order to avoid this, the adjoint method can be used in order to calculate the gradient at the cost of only one PDE solve [6, 7, 8]. The second difficulty is choosing the regularization parameter $\alpha \in \mathbb{R}^+$. Since this parameter models the balance between model fidelity ($\alpha \to 0$) and the regularity of k ($\alpha \to +\infty$), its value greatly influences the reconstruction. While many papers describe how the adjoint method can be used to solve nonlinear inverse problems, often the regularization parameter is chosen by trial-and-error or using the L-curve [9, 10, 11, 12]. Recent examples of this include [2, 8]. These approaches, however, require the solution of the inverse problem for many different values of the regularization parameter, which is inefficient, computationally expensive and may take a long time for large scale problems.

In this paper we derive two methods that simultaneously solve the inverse problem and determine a suitable value for the regularization parameter. The first method we call "generalized Newton-Tikhonov" (GNT) and is a direct generalization of the Generalized Arnoldi Tikhonov method (GAT) for linear inverse problems [13, 14, 15]. However, as we will demonstrate, GNT requires a number of redundant computations which were not needed in the original GAT method. The second method we call "regula falsi generalized Newton-Tikhonov" (RFGNT) and is a novel method based on similar ideas, but with a number of advantages which make it more efficient for nonlinear problems.

The outline of the paper is as follows. In section 2 we give an overview of how the regularization parameter can be chosen and how this is automatically

done by the generalized Arnoldi-Tikhonv method. This method will then be used as a basis for the GNT and RFGNT algorithms we derive in section 3. We then apply our methods to an inverse scattering problem in section 4, where we use the adjoint method for the gradient computations, and compare our results with other known regularization approaches.

2. Automatic regularization for linear problems

2.1. Choosing the regularization parameter

It is well known that when dealing with inverse problems and measured data some form of regularization is necessary in order to find a good solution [11, 12]. However, the regularization parameter α can greatly influence the outcome since it determines the balance between fitting the model to the noisy data and the regularization. If, on the one hand, α is too small, the regularization will have little to no effect and the noise in the data will corrupt the outcome of the algorithm. If, on the other hand, α is too large, this will lead to a solution that no longer fits the data very well. It may also have lost many small details and be what is referred to as "oversmoothed".

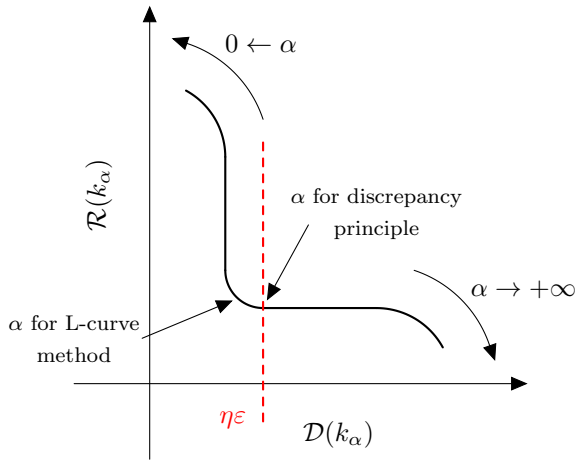

Figure 1: Sketch of the L-curve: the curve $(\mathcal{D}(k_\alpha), \mathcal{R}(k_\alpha))$ typically has a rough L-shape. The L-curve method proposes to use the regularization parameter which corresponds to the corner of the L. The discrepancy principle on the other hand uses the value that corresponds to the intersection of the curve and the vertical line at $\eta\varepsilon$. This value is typically slightly bigger [16]

Let k_α be the solution of (2) for a fixed regularization parameter α and $u(k_\alpha)$ the corresponding solution to the PDE (1). One way of determining a good value for α – and illustrating its effect on k – is the L-curve, see figure 1. By solving the inverse problem (2) for a whole range of values for α and looking at the the curve

$$(\mathcal{D}(k_\alpha), \mathcal{R}(k_\alpha)) = \left(\|Lu(k_\alpha) - \widetilde{u}\|^2, \|k_\alpha - k_0\|^2 \right),$$

it can be observed that it is roughly L-shaped. Heuristically, a "good" regularization parameter is the one that corresponds to the corner of the L, since this will balance model fidelity and regularization [11, 12].

Another way of choosing the regularization parameter α is the discrepancy principle [12, 13, 17], i.e. choose the regularization parameter such that

$$\mathcal{D}(\alpha) := \mathcal{D}(k_\alpha) = \eta \underbrace{\|Lu - \widetilde{u}\|^2}_{\varepsilon :=}.$$

Here, ε is called the error norm and $1 \leq \eta$ is a tolerance value. The motivation behind this choice is that decreasing the discrepancy $\mathcal{D}(\alpha)$ below the error norm will not necessarily improve the reconstruction and can lead to overfitting. The downside of the discrepancy principle is that (an estimate of) ε must be available.

2.2. Generalized Arnoldi-Tikhonov

The generalized Arnoldi-Tikhonov method was introduced in [13, 14, 15] as a method to solve the classical Tikhonov problem for linear problems of the form

$$\arg\min_{x \in \mathbb{R}^n} \|Ax - b\|^2 + \alpha \|x\|^2, \tag{3}$$

with $x, b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. It is an iterative algorithm that generates a sequence of approximations $x_0, x_1, x_2, \ldots$ that converge towards the solution of (3), while also updating the regularization parameter in each iteration. This is done based on the discrepancy principle and the current approximation of the solution. The method can best be understood by looking at the discrepancy curve $(\alpha, \mathcal{D}(\alpha))$, see figure 2, which can be seen as the analogue of the L-curve for the discrepancy principle.

The idea behind GAT is to use the secant method in order to approximate the value of α for which $\mathcal{D}(\alpha) = \eta\varepsilon$. The method is also a Krylov subspace method based on the Arnoldi decomposition of the matrix A [18, 19]. This means that the iterates for the solution of (3) are given by

$$x_{\alpha,i} := \arg\min_{x \in \mathcal{K}_i} \|Ax - b\|^2 + \alpha \|x\|^2,$$

with

$$\mathcal{K}_i = \mathcal{K}_i(A, b) = \operatorname{span}\left\{b, Ab, A^2b, \ldots, A^{i-1}b\right\}$$

the associated Krylov subspace of dimension i. In each iteration, a new basis vector is added to the Krylov subspace and the iterates are updated in order to account for this new basis vector. It is important to note that the constructed Krylov basis is independent of the regularization parameter. This means that it can be stored and reused in the next iteration when the regularization parameter is updated. In order to update the current best estimate for the regularization parameter α_{i-1}, GAT assumes that (3) is simultaneously solved without regularization, i.e. for $\alpha = 0$. This means that the points $(0, \mathcal{D}(0))$ and $(\alpha_{i-1}, \mathcal{D}(\alpha_{i-1}))$

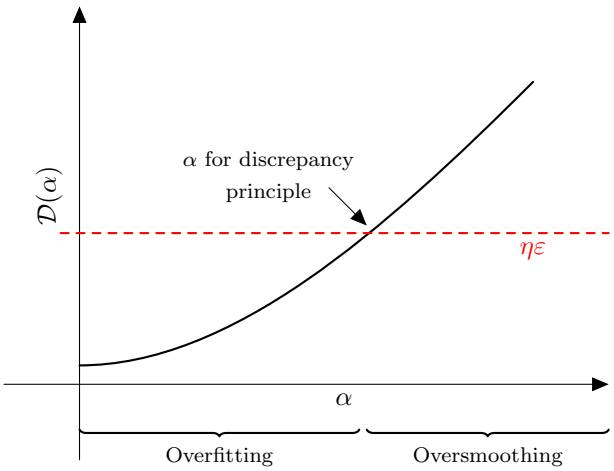

Figure 2: Plot of the discrepancy as a function of the regularization parameter. When α is small the model fidelity will be very high, but due to the noise in the data can exhibit overfitting. By increasing α more emphasis is put on the regularization term and overfitting is reduced. The discrepancy will start to increase however.

on the discrepancy curve are known and the regularization parameter can be updated using one step of the secant method:

$$\alpha_i = \frac{\eta\varepsilon - \mathcal{D}(0)}{\mathcal{D}(\alpha_{i-1}) - \mathcal{D}(0)}\alpha_{i-1}$$

Furthermore, instead of solving (3) to convergence each time the regularization parameter is updated and calculate the value $\mathcal{D}(\alpha_{i-1})$ exactly, the inverse problem is solved in the currently constructed Krylov subspace. This means that if

$$\mathcal{D}_i(\alpha) := \mathcal{D}(x_{\alpha,i}) = \|Ax_{\alpha,i} - b\|^2$$

is the discrepancy after i iterations – or equivalently the discrepancy in the Krylov subspace $\mathcal{K}_i$ of dimension i – then the GAT update for the regularization parameter is given by:

$$\alpha_i = \left|\frac{\eta\varepsilon - \mathcal{D}_i(0)}{\mathcal{D}_i(\alpha_{i-1}) - \mathcal{D}_i(0)}\right|\alpha_{i-1}. \tag{4}$$

Then, once the discrepancy principle is satisfied, i.e. $\mathcal{D}_i(\alpha_{i-1}) \leq \eta\varepsilon$, the algorithm is stopped.

Note that the absolute value is added because now it is possible for both $\mathcal{D}_i(0)$ and $\mathcal{D}_i(\alpha_{i-1})$ to be bigger than $\eta\varepsilon$, which can otherwise result in negative values for the regularization parameter. This only happens in the first few iterations when the constructed Krylov subspace is to small to contain a good approximation for the solution. Since the constructed Krylov basis is independent of α, the fact that the regularization parameter is estimated incorrectly in

the first few iterations does not matter. It is typically only when $\mathcal{D}_i(0)$ becomes smaller than $\eta\varepsilon$ that the estimates start to improve. Finally, we remark that the value $\alpha = 0$ is chosen and fixed for the secant updates because in this case the linear system that needs to be solved is smaller and the method becomes equivalent to the GMRES algorithm [20]. However, since the algorithm is stopped once the discrepancy principle is satisfied, this may result in an underestimation of the regularization parameter.

3. Automatic regularization for nonlinear problems

3.1. Generalized Newton-Tikhonov

Although GAT is a Krylov subspace method for linear the linear Tikhonov problem, the same idea can be used for nonlinear problems. The update for the regularization parameter (4) can even be applied directly if we change the notation back to our nonlinear problem. The only difference is how the iterates are calculated. If we use Newton's method to solve (2) and $k_{\alpha,i}$ is the ith Newton iteration using a fixed regularization parameter and $u(k_{\alpha,i})$ the corresponding solution to the PDE (1), then the discrepancy after i iterations is now given by

$$\mathcal{D}_i(\alpha) := \mathcal{D}(k_{\alpha,i}) = \|Lu(k_{\alpha,i}) - \widetilde{u}\|^2 . \tag{5}$$

An overview of this method, which we will call generalized Newton-Tikhonov (GNT), is given in algorithm 1. This method should be seen as a direct generalization of the GAT algorithm for nonlinear problems. As we will demonstrate with our numerical experiments, the method can be used to solve our nonlinear inverse problem, but it has a number of drawbacks that were not present in the original GAT method for linear inverse problems.

Algorithm 1 generalized Newton-Tikhonov (GNT)

Choose initial α_0, k_0
for $i = 1, \ldots,$ maxIter **do**
    Calculate $k_{0,i}$ based on $k_{0,i-1}$. ▷ This requires 1 Newton step.
    Calculate $k_{\alpha_{i-1},i}$ based on k_0. ▷ This requires i Newton steps.
    Calculate $\mathcal{D}_i(0)$ and $\mathcal{D}_i(\alpha_{i-1})$ using (5).
    if $\mathcal{D}_i(\alpha_{i-1}) \leq \eta\varepsilon$ **then**
        break
    else
        Calculate α_i using (4).
    end if
end for

A first thing to note is that both GAT and GNT update the regularization parameter based on the regularized and the non-regularized solution after a certain number of iterations. For the non-regularized iterations, this means that in each GNT iteration we need to perform one Newton step, see algorithm 1

line 3. However, when we wish to determine $k_{\alpha_{i-1},i}$, we cannot use previous best approximation $k_{\alpha_{i-2},i-1}$. This is because the Newton iterations depend on α, so we have to restart them from k_0 and perform i new Newton steps, see algorithm 1 line 4. In the original GAT method for linear problems this is not an issue, since the Krylov basis is independent from the regularization parameter. Therefore, in each iteration only one new Krylov basis vector has to be determined and added to the current basis. For GNT on the other hand, in each iteration the regularized Newton iterations have to be restarted and the number of Newton steps that needs to be computed increases each time. In order to to perform i GNT iterations, the number of Newton iterations needed is:

$$\underbrace{(1+1+\ldots+1)}_{i \text{ times line 3}}+\underbrace{(1+2+\ldots+i)}_{i \text{ times line 4}}=\frac{i(i+3)}{2}.$$

Another thing to remark is that in the original GAT method the reason for using $\alpha = 0$ as the second point for the secant step and not updating it, is because for this choice of α the linear system that needs to be solved is smaller and hence, easier to solve. To draw the parallel with the original method we also use this value for GNT. However, for nonlinear problems there is no direct benefit for using this value and if a better initial estimate for the regularization parameter is available or if α must be larger than 0 in order for the inverse problem to be well posed, another value can be used. This might also improve the quality of the estimate for the regularization parameter.

3.2. Regula falsi generalized Newton-Tikhonov

Two drawbacks of GNT are the increasing number of Newton iterations and the fact that the value for the regularization parameter only slowly converges to the value satisfying the discrepancy principle, which we will refer to as α^*. This can be seen in our numerical experiments, see figure 6, and is explained by the fact that the secant update step for the regularization parameter is done with a fixed point at $\alpha = 0$ and the discrepancy is only calculated up to a limited number of Newton iterations, see algorithm 1 line 9. Therefore, in order to limit the total number of Newton iterations and better approximate α^*, we propose an alternative approach based on the regula falsi method.

Assuming we have values $\alpha_0, \alpha_1 \in \mathbb{R}^+$ such that $\alpha_0 \leq \alpha^* \leq \alpha_1$, we determine the line between $(\alpha_0, \mathcal{D}(\alpha_0)$ and $(\alpha_1, \mathcal{D}(\alpha_1))$ and take α_2 as the value for which this equals the discrepancy:

$$\alpha_2 = \frac{\eta\varepsilon - \mathcal{D}(\alpha_0)}{\mathcal{D}(\alpha_1) - \mathcal{D}(\alpha_0)}(\alpha_1 - \alpha_0) + \alpha_0 \tag{6}$$

We then solve (2) with α_2 to determine $\mathcal{D}(\alpha_2)$ and replace either α_0 or α_1 with α_2, such that the interval $[\alpha_0, \alpha_1]$ will always contain α^*. Furthermore, in contrast to GNT, we will calculate $\mathcal{D}(\alpha)$ and k_α exactly and not up to a limited number of iterations. This is justified in the GAT algorithm, where due to the presence of a basis of the Krylov subspace the discrepancy after i

iterations could be acquired at a low cost. However, as we saw with GNT, this is no longer the case for nonlinear problems. In order to limit the total number of required Newton iterations we will instead update the initial guess for the Newton iterations every time α_2 is updated. Similarly to the regula falsi step for the regularization parameter, we take a weighted linear approximation based on the solutions in α_0 and α_1, i.e. k_{α_0} and k_{α_1}, which were already calculated:

$$k_{\alpha_2,0} = \frac{\alpha_2 - \alpha_0}{\alpha_1 - \alpha_0} \left(k_{\alpha_1} - k_{\alpha_0}\right) + k_{\alpha_0} \tag{7}$$

Then, once the value for α_2 starts to converge, we terminate the algorithm, see algorithm 2 line 9.

An overview of this algorithm, which we call regula falsi generalized Newton-Tikhonov (RFGNT), is given in algorithm 2. It should be noted that the only difference between the updates for the regularization parameter in GNT and in RFGNT is that in RFGNT α_0 does not have the fixed value 0. This also means that when the initial interval $[\alpha_0, \alpha_1]$ does not contain α^*, we can update the interval based on the secant method until it does, i.e. $\alpha_0 \leftarrow \alpha_1$ and $\alpha_1 \leftarrow \alpha_2$ (or vise versa if $\alpha_2 < \alpha_1$).

Algorithm 2 Regula falsi generalized Newton-Tikhonov (RFGNT)

1: Choose initial k_0.
2: Choose initial α_0 and α_1 such that $\alpha_0 \leq \alpha^* \leq \alpha_1$.
3: Calculate k_{α_0} and $\mathcal{D}(\alpha_0)$ by solving (2) (starting with initial k_0).
4: Calculate k_{α_1} and $\mathcal{D}(\alpha_1)$ by solving (2) (starting with initial k_0).
5: **for** $i = 1, \ldots,$ maxIter **do**
6: Calculate α_2 using (6).
7: Calculate $k_{\alpha_2,0}$ using (7).
8: Calculate k_{α_2} and $\mathcal{D}(\alpha_2)$ by solving (2) (starting with initial $k_{\alpha_2,0}$).
9: **if** $\left|\alpha_2 - \alpha_2^{old}\right| / \alpha_2^{old} < 10^{-3}$ **then**
10: **break**
11: **else**
12: $\alpha_2^{old} \leftarrow \alpha_2$
13: Replace α_0 or α_1 with α_2 based on $\mathcal{D}(\alpha_2)$.
14: **end if**
15: **end for**

4. Numerical experiments

4.1. Inverse scattering

Consider the homogeneous Helmholtz equation

$$\left(\Delta + k^2\right) u_{tot} = 0.$$

on a square domain $\Omega \subseteq \mathbb{R}^2$ with exterior complex scaling (ECS) boundary conditions and a spatially varying wave number $k : \Omega \longrightarrow \mathbb{R}$. We will assume

that the total wave $u_{tot} : \Omega \longrightarrow \mathbb{C}$ can be written as the sum of an incoming wave and the resulting scattered wave:

$$u_{tot} = u_{in} + u_{sc}.$$

If for multiple incoming waves of the form

$$u_{in}^{\theta}(x, y) = e^{ik_0(\cos\theta x + \sin\theta y)},$$

the resulting scattered waves are given by u_{sc}^{θ}, then this can be written as one big system of equations:

$$\underbrace{\begin{pmatrix} (\Delta + k^2) & & & \\ & (\Delta + k^2) & & \\ & & \ddots & \\ & & & (\Delta + k^2) \end{pmatrix}}_{\mathcal{H} :=} \begin{pmatrix} u_{in}^{\theta_1} + u_{sc}^{\theta_1} \\ u_{in}^{\theta_2} + u_{sc}^{\theta_2} \\ \vdots \\ u_{in}^{\theta_t} + u_{sc}^{\theta_t} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

If we denote $u_{in} = \left(u_{in}^{\theta_1}, \ldots, u_{in}^{\theta_t}\right)^T$ and $u_{sc} = \left(u_{sc}^{\theta_1}, \ldots, u_{sc}^{\theta_t}\right)^T$, then this is equivalent to

$$\mathcal{H} u_{sc} = \underbrace{\left(k_0^2 - k^2\right) u_{in}}_{\mathcal{F} :=}. \tag{8}$$

For our numerical experiment, we will try to reconstruct the wave number k based on measurements of the scattered wave at the boundary $\delta\Omega$. If $Hu = f$ is the discrete version of (8), $\widetilde{u}$ the measured values at $\delta\Omega$ and L the restriction operator that maps the full solution u on the discretized domain to its values on boundary of the domain, then we get the following constrained minimization problem:

$$\begin{cases} \min\limits_{k \in \mathbb{R}^n} \mathcal{J}(k) = \min\limits_{k \in \mathbb{R}^n} \|Lu - \widetilde{u}\|_2^2 + \alpha \|k - k_0\|_2^2 \\ \qquad Hu = f \end{cases} \tag{9}$$

4.2. The adjoint method

Because we will use Newton's method for the optimization in GNT and RFGNT, we calculate the gradient of $\mathcal{J}(k)$ using the adjoint method. Let $\langle x \mid y \rangle = x^* y$ denote the complex inner product for $x, y \in \mathbb{C}^n$ and x^* the conjugate transpose of x, then we can write the cost function as

$$\mathcal{J}(k) = \langle Lu - \widetilde{u} \,|Lu - \widetilde{u}\rangle + \alpha \langle k - k_0 \,|k - k_0 \rangle .$$

It follows that:

$$\begin{aligned} \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}k} &= \left\langle Lu - \widetilde{u} \middle| L\frac{\mathrm{d}u}{\mathrm{d}k} \right\rangle + \left\langle L\frac{\mathrm{d}u}{\mathrm{d}k} \middle| Lu - \widetilde{u} \right\rangle \\ &\quad + \alpha \langle 1 \,|k - k_0 \rangle + \alpha \langle k - k_0| \, 1 \rangle \\ &= 2Re\left(\left\langle L^*(Lu - \widetilde{u}) \middle| \frac{\mathrm{d}u}{\mathrm{d}k} \right\rangle + \alpha \langle k - k_0| \, 1 \rangle \right). \end{aligned}$$

The difficulty is now the derivative of the state variables with respect to the control variables, i.e. $\mathrm{d}u/\mathrm{d}k$. In order to avoid needing to calculate this term, the adjoint method introduces an adjoint variable $\lambda \in \mathbb{C}^m$ as the solution of

$$H^*(k)\lambda = L^*\left(Lu - \widetilde{u}\right), \tag{10}$$

where H was the matrix representing the discretized PDE, see (1). From (1) it also follows that

$$\frac{\mathrm{d}H}{\mathrm{d}k}u + H\frac{\mathrm{d}u}{\mathrm{d}k} = \frac{\mathrm{d}f}{\mathrm{d}k}.$$

Substituting this into the derivative of $\mathcal{J}$, we can eliminate the term $\mathrm{d}u/\mathrm{d}k$:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}k} &= 2Re\left(\left\langle H^*\lambda \middle| \frac{\mathrm{d}u}{\mathrm{d}k}\right\rangle + \alpha\left\langle k - k_0 \middle| 1\right\rangle\right) \\ &= 2Re\left(\left\langle \lambda \middle| H\frac{\mathrm{d}u}{\mathrm{d}k}\right\rangle + \alpha\left\langle k - k_0 \middle| 1\right\rangle\right) \\ &= 2Re\left(\left\langle \lambda \middle| \frac{\mathrm{d}f}{\mathrm{d}k} - \frac{\mathrm{d}H}{\mathrm{d}k}u\right\rangle + \alpha\left\langle k + k_0 \middle| 1\right\rangle\right).\end{aligned} \tag{11}$$

This means that the gradient of the cost function can now be evaluated in three steps:

i) Given k, solve the original PDE, see (1), for u.

ii) Use u to solve the adjoint PDE, see (10), for λ.

iii) Evaluate the gradient $\mathrm{d}\mathcal{J}/\mathrm{d}k$ using (11).

Note that solving the adjoint equation (10) is closely related to the original PDE (1) and that the cost of solving it will be similar. This also means that we can evaluate the gradient of $\mathcal{J}(k)$ at the cost of only one extra PDE solve per iteration. For more information on the adjoint method we refer to [6, 7].

4.3. Newton-Krylov

In order to solve the inverse problem (2), we will use an algorithm based on the line search Newton-CG method described in [5]. This is a Newton-Krylov method, where the parameters k are updated using Newton's method, i.e.

$$k_{\alpha,i+1} = k_{\alpha,i} + \gamma\Delta k,$$

and the Hessian system for the Newton search direction Δk is solved using a Krylov subspace method, CG in this case [5, 21]:

$$\nabla^2\mathcal{J}(k_{\alpha,i})\Delta k = -\nabla\mathcal{J}(k_{\alpha,i}).$$

The CG iterations are stopped once the residual is smaller than

$$\min\left(0.5, \sqrt{\|\nabla\mathcal{J}(k_{\alpha,i})\|}\right)\nabla\mathcal{J}(k_{\alpha,i})$$

or $\|\Delta k\| < 10^{-3}$. The line search will be a simple backtracking algorithm starting from $\gamma = 1$ and halving this value until

$$\mathcal{J}(k_{\alpha,i} + \gamma \Delta k) < \mathcal{J}(k_{\alpha,i}).$$

We terminate the Newton iterations once they start to stagnate, i.e.

$$\left| \frac{J(k_{\alpha,i+1}) - J(k_{\alpha,i})}{J(k_{\alpha,i})} \right| < 10^{-3}$$

It should be noted that each function evaluation comes at the cost of one PDE solve. Furthermore, each CG iteration requires one matrix vector product with the Hessian. In order to avoid this, a finite difference approximation can be used. Using a central difference scheme the approximation is given by:

$$\nabla^2 \mathcal{J}(k) v \approx \frac{\nabla \mathcal{J}(k + hv) - \nabla \mathcal{J}(k - hv)}{2h}. \tag{12}$$

This implies that each Newton iterations requires three gradient calculations. Using the adjoint method, this is equivalent to three solves of the original PDE and three solves of the adjoint PDE. By replacing the central difference approximation of the Hessian matrix-vector product (12) with a forward or backward difference approximation, this can be reduced to two. It should be noted that the Hessian matrix-vector product can also be determined using the second order adjoint method, see [6, 8, 22]. For simplicity we chose not to do so.

4.4. Discretization

The discretization of the Helmholtz systems in $\mathcal{H}$ is independent with respect to the angle of the incoming wave. H will therefore be a block diagonal matrix with each block being a discrete version of the operator $(\Delta + k^2)$. For our numerical experiment we take $\Omega = [-5, 5]^2$ and discretize it using a regular 200×200 grid with grid spacing h. We add a small buffer zone of 10 grid points before the points where the measurements are taken and then add another 10 grid points before the start of the complex tails for the exterior complex scaling.

The idea behind exterior complex scaling [23] is to extend the domain into the complex plane. By imposing Dirichlet boundary conditions at the end of these complex "tails", outgoing waves at the boundary $\delta\Omega$ can be simulated. Numerically we do this by adding points to the real domain (with the same spacing h) and rotating them into the complex plane under a chosen angle, see figure 3. We use 80 grid points for the complex tails – one third of the real domain – in each direction with an angle $\alpha = \pi/6$. This means that the full grid has size 400×400.

To discretize the Laplace operator we use second order finite differences, which on a regular grid with spacing h, is given by the formula:

$$\begin{aligned} \Delta u(x, y) \approx & \frac{u(x - h, y) - 2u(x, y) + u(x + h, y)}{h^2} \\ & + \frac{u(x, y - h) - 2u(x, y) + u(x, y + h)}{h^2} \end{aligned} \tag{13}$$

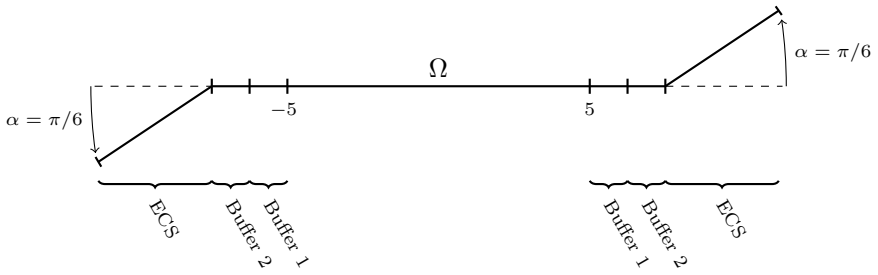

Figure 3: Exterior complex scaling in 1D: by adding complex tails to the domain and Dirichlet boundary conditions at the end of these tails, outgoing waves at the boundary $\partial\Omega$ can be simulated. The angle with respect to the real axis and how far the complex tails extend into the complex plain can be chosen.

However, because of the way the complex tails are constructed, the full grid is no longer regular. Therefore, the general form for irregular grids has to be used. Furthermore, we will assume that the wave number $k \in \mathbb{R}^n$ with $n = 200^2$ is equal to a base value $k_0 = 1$ outside of Ω. Inside the region of interest itself we add an offset based on the sum of three Gaussian functions placed symmetrically on a circle with radius $r = 2.5$:

$$k = k_0\sqrt{1+\chi},$$

with

$$\begin{aligned}\chi(x,y) = e^{-(x-r)^2-y^2} &+ e^{-\left(x-r\cos\left(\frac{2\pi}{3}\right)\right)^2-\left(y-r\sin\left(\frac{2\pi}{3}\right)\right)^2} \\ &+ e^{-\left(x-r\cos\left(\frac{4\pi}{3}\right)\right)^2-\left(y-r\sin\left(\frac{4\pi}{3}\right)\right)^2}.\end{aligned}$$

In order to simulate the measurements, we generate incoming waves of the form

$$u_{in}^{\theta}(x,y) = e^{ik_0(\cos\theta x+\sin\theta y)}$$

for 50 different value of $\theta \in [0, 2\pi[$ and calculate the corresponding scattered waves $u_{exact} \in \mathbb{C}^{50\cdot 400^2}$. Taking into account the 10 buffer points we added, we have 876 observations for every angle θ, which we select using the matrix $L \in \mathbb{R}^{50\cdot 876\times 50\cdot 400^2}$, see figure 4. We then add random Gaussian white noise to generate the measurements

$$\widetilde{u} = Lu_{exact} + \sigma\,(e_1 + ie_2)$$

with e_1 and $e_2 \sim \mathcal{N}(0, I_{50\cdot 876})$ and choose σ such that the noise level is approximately 10%:

$$\frac{\|Lu_{exact} - \widetilde{u}\|^2}{\|Lu_{exact}\|^2} \approx 0.10$$

This resulted in a value $\sigma = 0.075$ and $\varepsilon = 492.7031$, which we use for the discrepancy principle combined with $\eta = 1$. Because of the diagonal structure

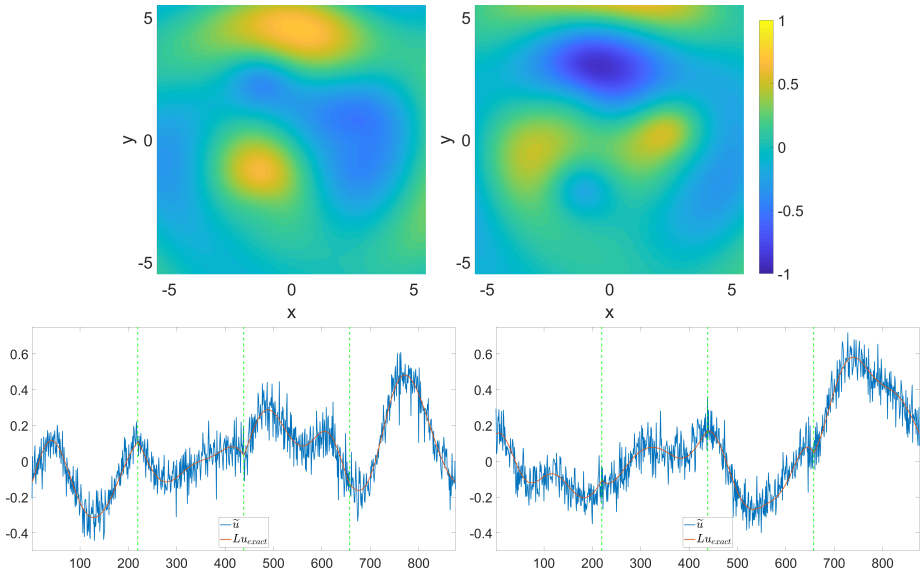

Figure 4: Top: real and complex part of the scattered wave for $\theta = 86.4°$ (left and right) on Ω, including the 10 buffer points to where the measurements are taken. Bottom: real and complex values of the same scattered wave (left and right) at the measurement points. The measurements are ordered counter clockwise starting from the top left corner of the domain and the dashed lines correspond to the corners.

of (8) we also do not construct the matrix H explicitly, but rather solve the 50 Helmholtz systems for the different angles in parallel, where each discrete version of the Helmholtz system $(\Delta + k^2)$ has size 160000×160000.

4.5. Results

In order to test the GNT and RFGNT algorithms, we consider two other reconstruction methods. The first is Newton's method without a regularization term, but with the discrepancy as an early stopping criterion. The second is the L-curve approach. We calculated 26 points on the L-curve for $\alpha \in [0, 0.5]$ and selected the one with the smallest error with respect to the exact wave number. Since the D-curve is another way of looking at the L-curve, we can also use these points to see whether or not the regularization parameter found by GNT and RFGNT is correct. Also, for GNT and RFGNT we solved the problem using a backward (B), a central (C) and a forward (F) finite difference approximation for the Hessian matrix vector product in the CG iterations. However, since there was little difference in the quality of the reconstructions, the early stopping solution and L-curve solution were calculated only for the forward finite difference approximation. The details of all the reconstructions are listed in table 1 and figures 5 and 6 illustrate some of these results for the forward finite difference scheme.

When comparing the different reconstructions, we see that there is little difference between the finite difference schemes, except in the number of gradient

		Newton iterations	CG iterations	$\mathcal{J}$ evaluations	$\nabla\mathcal{J}$ evaluations	PDE solves	Adjoint solves	$(\Delta + k^2)$ solves	Relative error	α
B	GNT	54	175	101	108	209	108	15850	0.0803	0.3431
	RFGNT	21	73	44	42	86	42	6400	0.0746	0.3613
C	GNT	65	225	118	195	313	195	25400	0.0702	0.3398
	RFGNT	21	72	44	63	107	63	8500	0.0746	0.3613
F	GNT	54	175	101	108	209	108	15850	0.0804	0.3429
	RFGNT	21	81	44	42	86	42	6400	0.0747	0.3613
	Early stopping	3	7	8	6	14	6	1000	0.2089	·
	L-curve	9	37	20	18	38	18	2800	0.0871	0.1400
	L-curve (all)	267	983	633	534	1167	534	85050	·	·

Table 1: Details from the different reconstructions for the different reconstruction methods and finite difference approximations for the Hessian matrix vector product in the CG iterations. The number of gradient evaluations is 2 or 3 times the number of Newton iterations (depending on the finite difference scheme) which is equal to the number of solves of the adjoint PDE. The number of PDE solves is sum of the number of cost function evaluations and the number of gradient evaluations. The number of Helmholtz system solves is 50 times, i.e. the number of projection angles, the number of PDE and Adjoint solves.

evaluations (and hence the number of PDE, adjoint PDE and Helmholtz solves). We also see that while GNT finds a good solution for the inverse problem, it requires a lot more Newton iterations to do so. This can also be seen when looking at the relative error and regularization parameter in figure 6. Each time the Newton method is restarted, progress is lost. While this approach was natural for linear problems combined with a Krylov subspace method, it is inefficient for nonlinear problems. The adaptations we made in order to derive the RFGNT method on the other hand seem to be very effective. The method first needs to solve the problem for our initial choices for $\alpha_0 = 0$ and $\alpha_1 = 1$, but due to the update of the initial estimate for the Newton iterations, only 1 or 2 Newton iterations are needed afterwards for line 8 of algorithm 2. The convergence of the regularization parameter is also drastically increased and by looking at the discrepancy curve in figure 6, we see that the method does indeed converge to the desired value for the regularization parameter.

When we compare GNT and RFGNT with early stopping, we see that the latter needs less iterations before satisfying its stopping criterion. Then again, the early stopping reconstruction has a much larger relative error than the other the reconstructions. Using the L-curve approach, on the other hand, we find a reconstruction that has the same quality as the GNT and RFGNT reconstructions. However, while calculating a single point on the L-curve is cheaper than solving the inverse problem with GNT or RFGNT, calculating all points the L-curve (26 in this case) is much less efficient. This clearly illustrates the effectiveness off the automatic regularization approach of RFGNT.

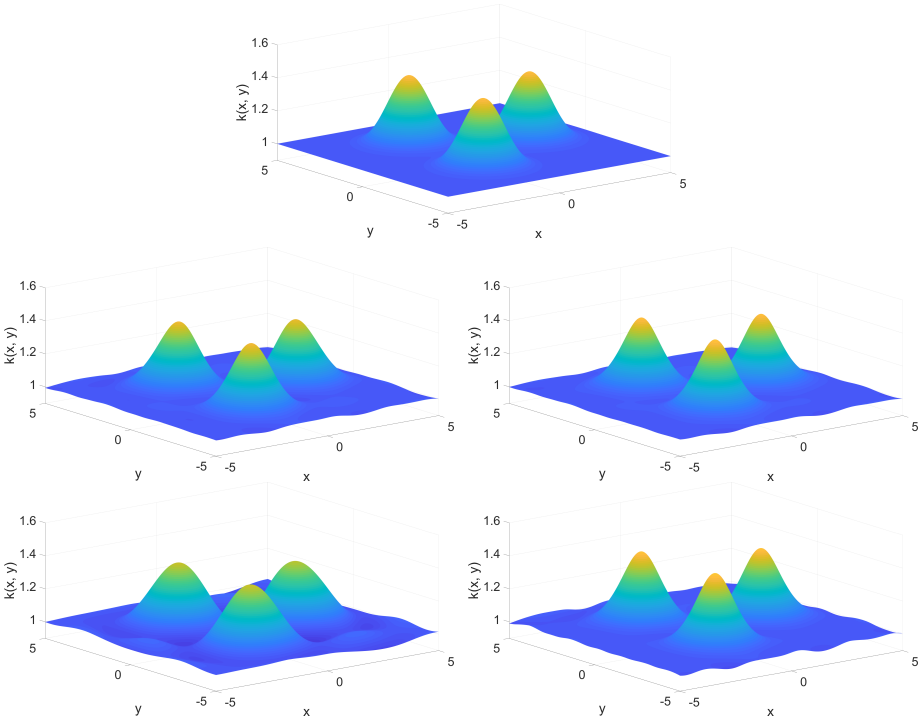

Figure 5: Top: the exact wave number k. Middle left: GNT reconstruction. Middle right: RFGNT reconstruction. Bottom left: early stopping reconstruction. Bottom right: L-curve reconstruction.

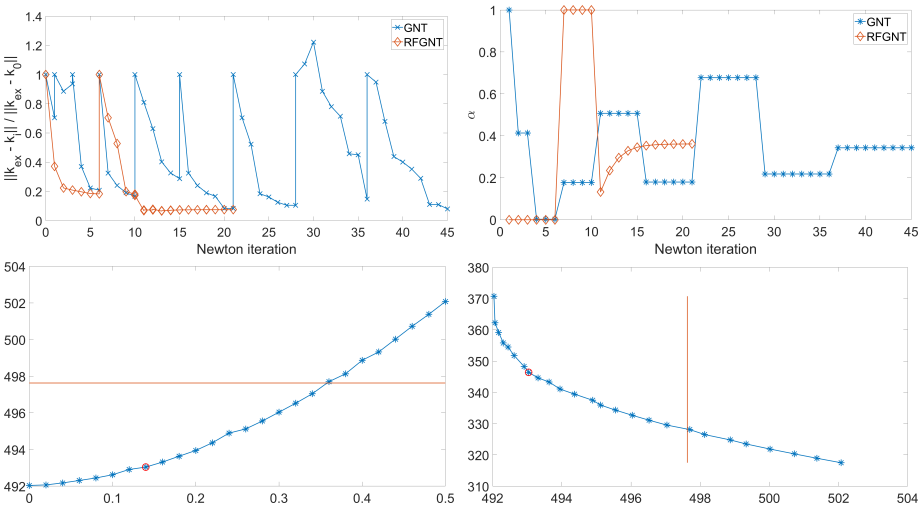

Figure 6: Top left: the relative error of GNT and RFGNT in each Newton iteration. We can clearly see at which point GNT restarts its Newton iterations. Top right: The regularization parameter used in each Newton iteration. Bottom: D-curve and L-curve. The red line corresponds to the discrepancy value $\eta\varepsilon$ and the point with the lowest error w.r.t. the exact solution is marked with a red circle.

5. Conclusion and remarks

In this paper we describe two methods to solve a nonlinear inverse problem that iteratively determine the solution and the regularization parameter. The first method, generalized Newton-Tikhonov (GNT), is a direct generalization of the generalized Arnoldi-Tikhonov method to nonlinear problems. However, this method turns out to have a number of drawbacks that were not present in the original algorithm for linear problems. In order to improve the method, we proposed the regula falsi generalized Newton-Tikhonov method (RFGNT). We replace the secant update step from GNT with a regula falsi approach and updating the initial guess for the Newton iterations with every update of the regularization parameter. This decreases the number of Newton iterations needed and finds a better value for the regularization parameter. Our numerical experiments also show that this is computationally much more efficient than, for example, calculating the L-curve or other grid based approaches to determine the regularization parameter.

It should also be noted that in this paper we solve the PDE and the adjoint PDE sequentially. By contrast, it is also possible to solve both simultaneously by considering the Karush-Kuhn-Tucker conditions and using Newton's method combined with a suitable preconditioner to find the optimum, see for example [24, 25, 26, 27]. The difficulty with this approach is finding a suitable preconditioner for the problem, but it can easily be combined with the proposed RFGNT method. This is because using this approach to solve the problem for a fixed value of the regularization parameter α corresponds to calculating k_α and $\mathcal{D}(\alpha)$,

i.e. lines 3, 4 and 8 of algorithm 2.

Furthermore, although we used an inverse scattering problem as a test problem to demonstrate the methods, no specific properties of this problem were used to derive the methods and they are likely to be effective in many other inverse problems. Future work therefore includes more in-depth analysis of the robustness of the methods and its use in other application. We did, for example, not use any preconditioner for the solution of the Helmholtz problem or the inner CG iterations. However, there has been done many interesting work concerning preconditioners for the Helmholtz equations using multigrid methods [28, 29]. Including these in the optimization procedure could reduce the number of required solves of the Helmholtz equation and CG iterations. Another possible issue is the fact that since Newton can only be used for local minimization a proper initial estimate needs to be determined.

Acknowledgement

The authors wish to thank the Department of Mathematics and Computer Science, University of Antwerp, for financial support.

References

References

[1] G. S. Abdoulaev, K. Ren, A. H. Hielscher, Optical tomography as a PDE-constrained optimization problem, Inverse Problems 21 (5) (2005) 1507.

[2] F. Bruckner, C. Abert, G. Wautischer, C. Huber, C. Vogler, M. Hinze, D. Suess, Solving large-scale inverse magnetostatic problems using the adjoint method, Scientific reports 7 (2017) 40816.

[3] C. Kaebe, J. H. Maruhn, E. W. Sachs, Adjoint-based Monte Carlo calibration of financial market models, Finance and Stochastics 13 (3) (2009) 351–379.

[4] K. J. In't Hout, S. Foulon, ADI finite difference schemes for option pricing in the Heston model with correlation, International journal of numerical analysis and modeling 7 (2) (2010) 303–320.

[5] J. Nocedal, S. J. Wright, Numerical optimization, Springer, 2006.

[6] D. A. Tortorelli, P. Michaleris, Design sensitivity analysis: overview and review, Inverse problems in Engineering 1 (1) (1994) 71–105.

[7] R. Plessix, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, Geophysical Journal International 167 (2) (2006) 495–503.

[8] B. Jadamba, A. A. Khan, A. A. Oberai, M. Sama, First-order and second-order adjoint methods for parameter identification problems with an application to the elasticity imaging inverse problem, Inverse Problems in Science and Engineering 25 (12) (2017) 1768–1787.

[9] D. Calvetti, G. H. Golub, L. Reichel, Estimation of the L-curve via Lanczos bidiagonalization, BIT Numerical Mathematics 39 (4) (1999) 603–619.

[10] D. Calvetti, L. Reichel, A. Shuibi, L-curve and curvature bounds for tikhonov regularization, Numerical Algorithms 35 (2–4) (2004) 301–314.

[11] P. C. Hansen, Discrete inverse problems: insight and algorithms, Vol. 7, Siam, 2010.

[12] C. R. Vogel, Computational methods for inverse problems, SIAM, 2002.

[13] S. Gazzola, J. G. Nagy, Generalized Arnoldi-Tikhonov method for sparse reconstruction, SIAM Journal on Scientific Computing 36 (2) (2014) B225–B247.

[14] S. Gazzola, P. Novati, Automatic parameter setting for Arnoldi-Tikhonov methods, Journal of Computational and Applied Mathematics 256 (2014) 180–195.

[15] S. Gazzola, P. Novati, M. R. Russo, On krylov projection methods and tikhonov regularization, Electron. Trans. Numer. Anal 44 (2015) 83–123.

[16] P. C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, SIAM review 34 (4) (1992) 561–580.

[17] V. A. Morozov, Methods for Solving Incorrectly Posed Problems, Springer-Verslag, 1984.

[18] Y. Saad, Iterative methods for sparse linear systems, Vol. 82, siam, 2003.

[19] H. A. Van der Vorst, Iterative Krylov methods for large linear systems, Vol. 13, Cambridge University Press, 2003.

[20] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.

[21] J. R. Shewchuk, et al., An introduction to the conjugate gradient method without the agonizing pain (1994).

[22] Z. Wang, I. M. Navon, F. X. L. Dimet, X. Zou, The second order adjoint analysis: theory and applications, Meteorology and atmospheric physics 50 (1-3) (1992) 3–20.

[23] C. W. McCurdy, M. Baertschy, T. N. Rescigno, Solving the three-body coulomb breakup problem using exterior complex scaling, Journal of Physics B: Atomic, Molecular and Optical Physics 37 (17) (2004) R137.

[24] G. Biros, O. Ghattas, Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. part I: The Krylov–Schur solver, SIAM Journal on Scientific Computing 27 (2) (2005) 687–713.

[25] E. Haber, U. M. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse Problems 16 (5) (2000) 1263.

[26] E. Haber, U. M. Ascher, Preconditioned all-at-once methods for large, sparse parameter estimation problems, Inverse Problems 17 (6) (2001) 1847.

[27] K.-A. Mardal, B. F. Nielsen, M. Nordaas, Robust preconditioners for PDE-constrained optimization with limited observations, BIT Numerical Mathematics 57 (2) (2017) 405–431.

[28] S. Cools, W. Vanroose, Local Fourier analysis of the complex shifted Laplacian preconditioner for Helmholtz problems, Numerical Linear Algebra with Applications 20 (4) (2013) 575–597.

[29] Y. A. Erlangga, C. Vuik, C. W. Oosterlee, Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation, Applied numerical mathematics 56 (5) (2006) 648–666.