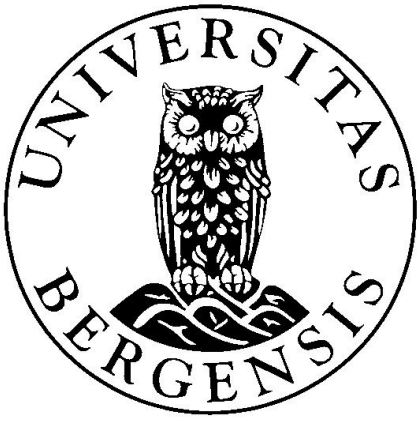
Gradient Methods for Source Identification Problems

Shin Irgens Banshoya

MASTER THESIS IN APPLIED AND COMPUTATIONAL MATHEMATICS



Department of Mathematics

University of Bergen

June 2020

Acknowledgement

First, I want to thank my two supervisors, Guttorm Alendal and Anna Oleynik, for all help they have given me.
I also want to thank my friends at the university making my life enjoyable.
Finally, i want to thank my family for all the support they have given me over the years.

Abstract

In this thesis we are interested in the problem of identifying point sources in an advection-diffusion model, given some measurements. By writing the problem as a minimization problem, we use a non-linear conjugate gradient method in order to estimate the source location and the corresponding intensity. Several test cases are presented. The primary motivation for this work is monitoring CO_2 storage at deep geological formations, where solid monitoring tools are imperative for the storage in order to be considered safe. However, this can also be used in other situations when estimating the source location of some contamination is of interest.

Contents

Chapter 1

Introduction

Due to the increase in atmospheric carbon dioxide (CO_2) concentration, the earth's environment has changed. This includes global warming and ocean acidification ([29]), which have caused some appreciable events, like extreme weather events ([5]). Preventing such events from worsening and other occurences from happening, such as losses of croral reefs ([23]), reducing CO_2 emission is essential.

One promising posibility in order to solve this problem, is Carbon dioxide Capture and Storage (CCS), which refers to CO_2 emission redusing technologies ([18]). In CCS, CO_2 is captured, for example, from large industrial power plants and stored in deep geological formations. In some countries, e.g. Japan, such formations are located deep under the seabed ([29]).

There is a risk with CCS of leakages from a storage site ([9]). If a leak happens, the question is what impact it will have on the marine environment. One of the main factors here is the deacrease of pH value ([9]). In [9], it is stated that if the decrease is less than 0.5 pH units, then the impact is minimal. A larger decrease may also have a minimal impact, but in short term. In long term, on the other hand, a large decrease in pH has a large

impact. If the decrease is above 1 pH units, then there is a large decrease in biomass, and this could lead to biomass loss for the macrofauna ([9]). Other factors, such as salinity and natural seasonal events will also have an impact on the effect of the leakage ([29, 9]). Therefore monitoring both the reservoir and its surroundings is important, in order to ensure that the CO_2 is safely stored ([29]). The problem in localizing the source of a leak is an application of a problem that some authors (e.g. [21]) call a *source identification problem*.

Source identification problems appares in many branches of science (see [19] for a list of examples). For example, the authors of [19] studie an iterative procedure for the Poisson equation with measurements taken at the boundary. In [21], the authors propose an iterative method for systems of advection-diffusion-reaction equations that occur in the atmosphere.

In this thesis we use gradient methods for the source identification problem. By writing the problem as a PDE-constrained optimization problem, we minimize an objective function that measures the error between measurements and a model. This could be used to reveal, for example, the origin of a CO_2 leak. In this study we work in $\mathbb{R}^2$, but extending the results to $\mathbb{R}^3$ is rather straightforward.

In Chapter 2 we first give a mathematical formulation of the problem and discuss some necessary mathematical theory. The numerical methods are discussed in Chapter 3. In Chapter 4 and 5 we present results from different test cases. Finally, we give some concluding remarks in Chapter 6.

Chapter 2

Mathematical framework

In this chapter, we discuss the mathematical framework to solve the minimization problem. In the first section we give a mathematical formulation of the problem, while in the two next sections we discuss some mathematical conditions related to existence of a solution.

2.1 Formulation of the problem

Let $\Omega \subset \mathbb{R}^2$ be a bounded and open domain with boundary $\partial\Omega$ and let u be a concentration of some pollutant, e.g. CO_2. The evolution of the concentration is modeled by an advection-diffusion initial-boundary-value problem (IBVP) ([2, 15]),

$$\frac{\partial u}{\partial t} - D\Delta u + \mathbf{w} \cdot \nabla u = F(\mathbf{x}, t), \quad \mathbf{x} \in \Omega,\, t \geq 0 \tag{2.1}$$

$$u(\mathbf{x}, 0) = 0 \tag{2.2}$$

$$u|_{\partial\Omega} = 0, \tag{2.3}$$

where F is a source function. Here, $\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$ is a divergence free velocity field, D is a diffusion coefficient, and ∇ and Δ denote the nabla and Laplace operators, respectively.

While the initial condition is set to zero and we use homogeneous Dirichlet boundary condition, note that the IBVP is linear. Hence we could always add a solution with either a non-zero initial condition or with inhomogeneous boundary conditions to this problem, and it will not alter the results ([15]). See [15] for a discussion of initial and boudary conditions for a similar IBVP.

For source identification problems, the source function is on the form ([15, 21])

$$F = \sum_{i=1}^{N_s} a_i \delta(\mathbf{x} - \boldsymbol{\xi}_i), \tag{2.4}$$

where $\boldsymbol{\xi}_i$ is the position of source number i, and a_i is the corresponding source intensity. The total number of sources is denoted N_s. $\delta(\mathbf{x})$ is the 2-dimensional Dirac delta-distribution. In [19], however, it is argued that the source function can be written

$$F = \sum_{i=1}^{N_s} a_i \Phi(\|\mathbf{x} - \boldsymbol{\xi}_i\|_2; \epsilon_i), \tag{2.5}$$

where, and for the rest of the thesis, $\|\cdot\|_2$ denotes the Euclidean norm, and $\Phi(\mathbf{x}; \epsilon)$ is a cylindrical function, defined as

$$\Phi(\mathbf{x}; \epsilon) = \begin{cases} 1 & 0 \leq \|\mathbf{x}\|_2 < \epsilon, \\ 0 & \|\mathbf{x}\|_2 \geq \epsilon. \end{cases}$$

Given a source function, the *forward problem* is to compute a numerical solution u. We are interested in *the inverse problem* of estimating the parameters in the source function F. That is, find the source positions and the corresponding intensities such that the distance between measurements,

denoted $u_m^*(t)$, and a numerical solution u is as small as possible. The numerical solution is computed based on the estimated parameters, and the measurements are taken at position $\mathbf{x}_m \in \Omega$, $m = 1, \ldots, N_m$.

To be more precise, let $\mathbf{d}$ be a vector containing all the parameters $\{a_i, \boldsymbol{\xi_i}\}$, $i = 1, 2, \ldots, N_s$. In order to do the estimation, we want to minimize the (objective) function

$$J(u, \mathbf{d}) = \frac{1}{2} \sum_{m=1}^{N_m} \int_0^T \frac{(u(\mathbf{x}_m, t; \mathbf{d}) - u_m^*(t))^2}{w_m^2} dt, \tag{2.6}$$

with respect to $\mathbf{d}$. $u(\mathbf{x}, t; \mathbf{d})$ is the solution to the advection-diffusion model (2.1)-(2.3) and w_m are given weights.

2.2 Well-posedness

When solving a problem, it is important to know whether the problem has a solution, if the solution is unique, and if it depends continuously on data. If a solution satisfies these three conditions, then the problem is well-posed ([2, 28]).

For instance, the IBVP (2.1)-(2.3), with the source function (2.5) is well-posed in the space $L^2(\Omega)$ (see Appendix C for a definition of this space). Indeed, multiply the equation by u and integrate over space (known as the energy method),

$$\iint_\Omega u u_t - Du\Delta u + u(\mathbf{w} \cdot \nabla u)\, d\mathbf{x} = \iint_\Omega uF d\mathbf{x},$$

where $d\mathbf{x} = dxdy$. The first term on the left-hand-side can be written as

$$\iint_\Omega u u_t \, d\mathbf{x} = \frac{1}{2} \iint_\Omega \frac{\partial (u)^2}{\partial t} \, d\mathbf{x} = \frac{1}{2} \frac{d}{dt} \|u\|_2^2,$$

where we made use of the fact that the Euclidean norm is induced by the inner product $(u, v) = \iint_\Omega uv d\mathbf{x}$, $\|u\|_2^2 = (u, u)$. For the two next terms we apply Green's first identity (see Appendix B) and integration by parts, which yield

$$\iint_\Omega u\Delta u d\mathbf{x} = \iint_\Omega -\nabla u \cdot \nabla u \, d\mathbf{x} + \int_{\partial\Omega} u(\nabla u \cdot \mathbf{n}) \, ds,$$
$$\iint_\Omega u(\mathbf{w} \cdot \nabla u) d\mathbf{x} = \int_{\partial\Omega} u \, u \, (\mathbf{w} \cdot \mathbf{n}) \, ds - \iint_\Omega u(\nabla \cdot (u\mathbf{w})) \, d\mathbf{x}.$$

($\mathbf{n} \cdot \nabla u$ is the directional derivative in the outward normal direction and ds is the element of arc length). After using the identity $(B.1)$, the boundary condition (2.3), and that the velocity field is divergence free, we obtain

$$\iint_\Omega u\Delta u d\mathbf{x} = -\|\nabla u\|_2^2,$$
$$2 \iint_\Omega u(\mathbf{w} \cdot \nabla u) \, dx = 0.$$

For the integral on the right-hand-side we have

$$\iint_\Omega uF d\mathbf{x} \leq \iint_\Omega |u \, F| d\mathbf{x} \leq \|u\|_2 \, \|F\|_2 \leq \frac{1}{2}\|u\|_2^2 + \frac{1}{2}\|F\|_2^2.$$

by the Cauchy-Schwarz and Cauchy inequalities ([8]). Thus

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}\|u\|_2^2 &= -D\|\nabla u\|_2^2 + \iint_\Omega uF d\mathbf{x} \\ &\leq \frac{1}{2}\|u\|_2^2 + \frac{1}{2}\|F\|_2^2. \end{aligned}$$

By Gronwall's inequality on differential form ([8]),

$$\|u(\mathbf{x}, T)\|_2^2 \leq e^T \int_0^T \|F(\mathbf{x}, t)\|_2^2 dt,$$

so we have well-posedness, as an IBVP on the form

$$\begin{aligned} u_t &= P(\mathbf{x},\, t,\, \partial_{\mathbf{x}})u + F(\mathbf{x}, t), \quad \mathbf{x} \in \Omega,\, t \geq 0 \\ u(\mathbf{x}, 0) &= g(\mathbf{x}), \\ L(t, \partial_{\mathbf{x}})u(\cdot,\, t)|_{\partial\Omega} &= 0, \end{aligned}$$

where P is a differential operator and L is a operator defining the boundary conditions, is well-posed if we have the bound

$$\|u(\mathbf{x}, t)\|_2^2 \leq K(t)\left(\|g(\mathbf{x})\|_2^2 + \int_0^t \|F(\mathbf{x}, t)\|_2^2 dt \right)$$

for a function K independent of g and F ([12, 13]).

As the Dirac delta-distribution is not in L^2, the above estimate do not hold for the source function (2.4). In this situation, existence and uniquness of a solution could be obtained in the context of *source-type solutions* ([21]). See [20] for a definition of a source-type solution. However, to the author's knowledge, there are no articles where such a result has been established for the problem (2.1)-(2.3), with (2.4).

In any case, it is stated in [15] that the IBVP (2.1)-(2.3) with (2.4) does have a unique solution in the space

$$L^2(0, T; L^2(\Omega)) \cap C(0, T; H^{-1}(\Omega)).$$

The definitions of the these function spaces is given in Appendix C.

For an inverse problem, there might be no solution, i.e. parameters, and therefore a mathematical model might not exactly fit the given measurements ([2]). Possible reasons for this are that the inverse problem is an overdetermined system of equations ([27]), or that measurements contain noise ([2]).

Several models, on the other hand, might satisfy the measurements. An

example could be that the inverse problem is a linear system of equations with more unknowns than equations ([2]).

Finally, the computations of a solution to an inverse problem can be unstable ([2, 27]). The instability is related to sensitivity of small changes, such as rounding error or noise in the measurements, which may lead to huge errors in the computed solution ([2, 27]).

Tikhonov regularization is one simple tecnique to stabilize the computations of a solution to an inverse problem ([2]). With this technique, we assume that there are many solutions that fit the measurements, such that the residual of the problem is smaller than some ν. Among these solutions, we select the one with the smallest norm, i.e.

$$\begin{aligned} &\min \|\mathbf{d}\|_2 \\ &\|J(u,\mathbf{d}) - \mathbf{b}\|_2 < \nu \end{aligned} \tag{2.7}$$

where $\mathbf{d}$ is the solution and $\mathbf{b}$ is a vector containing the measurements.

The motivation of mimimizing the norm of $\mathbf{d}$, is to find a solution that both have sufficient information of the problem and a small residual ([2]).

Using the method of Lagrange multipliers (section 2.3), the problem (2.7) can be rewritten as

$$\min_{\mathbf{d}} \; J(u,\mathbf{d}) + \tau\|\mathbf{d}\|_2^2,$$

where $\tau > 0$ is a regularization parameter. The solution of this problem is a regularized solution, that may however loose fit to the measurements, but gains solution stability ([2]).

2.3 Lagrange multipliers and KKT conditions

We want to solve

$$\min_d J(u, \mathbf{d}),$$

where J is given by (2.6), subject to u being the solution to

$$\begin{aligned}
&\frac{\partial u}{\partial t} - D\Delta u + \mathbf{w} \cdot \nabla u = F(\mathbf{x}, t), \quad \mathbf{x} \in \Omega,\ t \geq 0 \\
&u(\mathbf{x}, 0) = 0, \\
&u|_{\partial\Omega} = 0.
\end{aligned}$$

This is a constrained optimization problem, and can be solved by the method of Lagrange multipliers ([1, 2]). The idea of this method is to write the constrained optimization problem as a function, called the Lagrange function. Then a solution of the optimization problem can be found by looking for critical points (points where the gradient vanishes) of this function. The Lagrange function for the objective function (2.6), subject to the IBVP (2.1)-(2.3) reads ([24])

$$\begin{aligned}
L(u, \mathbf{d}, \lambda) = \frac{1}{2} \sum_{m=1}^{N_m} \int_{t=0}^{t=T} \iint_\Omega \frac{(u(\mathbf{x}, t; \mathbf{d}) - u_m^*(t)\,)^2 \delta(\mathbf{x} - \mathbf{x}_m)}{w_m^2} d\mathbf{x} dt \qquad (2.8) \\
+ \int_{t=0}^{t=T} \iint_\Omega \lambda \Big(u_t - D\Delta u + \mathbf{w} \cdot \nabla u - F \Big) d\mathbf{x} dt.
\end{aligned}$$

Here, $\lambda = \lambda(\mathbf{x}, t)$ is called a Lagrange multiplier. Note also that we have used the identity $g(\mathbf{x}_0) = \iint_\Omega g(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}_0) d\mathbf{x}$ on the objective function (2.6).

However, the method does not guarantee that a solution exists. In order to know whether a solution, $\mathbf{d}$, exists or not, we need the Karuch-Kuhn-Tucker (KKT) conditions. These are necessary conditions for existence of a solution ([3, 25]). The following theorem and two definitions (from [3])

explain these conditions. Note that the constraints here are functions, not PDEs. For the case of PDE-constraints, we refer to [10, 16].

Theorem 2.1. *Let $\boldsymbol{x}_*$ be a (local) solution of*

$$\begin{aligned} &\min_x f(\boldsymbol{x}) \\ \text{subject to } &h_j(\boldsymbol{x}) = 0, \quad j = 1, 2, ..., M_1. \\ &g_i(\boldsymbol{x}) \leq 0, \quad i = 1, 2, ..., M_2 \end{aligned} \tag{2.9}$$

where f, h_j and g_i are all continuous differentiable functions over $\mathbb{R}^d$. Suppose that the gradients of the active constraints (i.e. $I(\boldsymbol{x}_*) = \{i : g_i(\boldsymbol{x}_*) = 0\}$) and the equality constraints

$$\{\nabla g_i(\boldsymbol{x}_*) : i \in I(\boldsymbol{x}_*)\} \cup \{\nabla h_j(\boldsymbol{x}_*)\}$$

are linear independent. Then there exist multipliers, called Lagrange multipliers, $\lambda_j \in \mathbb{R}$ and $\mu_i \geq 0$ such that

$$\begin{aligned} \nabla f(\boldsymbol{x}_*) + \sum_{j=1}^{M_1} \lambda_j \nabla h_j(\boldsymbol{x}_*) + \sum_{i=1}^{M_2} \mu_i \nabla g_i(\boldsymbol{x}_*) = \boldsymbol{0}, \\ \mu_i g_i(\boldsymbol{x}_*) = 0, \quad i = 1,\, 2,\, \ldots,\, M_2. \end{aligned}$$

□

Definition 2.1 (**KKT point**). *Consider the minimization problem* (2.9). *A solution $\boldsymbol{x}_*$ is called a KKT point if there exist $\lambda_j \in \mathbb{R}$ and $\mu_i \geq 0$ such that*

$$\begin{aligned} \nabla f(\boldsymbol{x}_*) + \sum_{j=1}^{M_1} \lambda_j \nabla h_j(\boldsymbol{x}_*) + \sum_{i=1}^{M_2} \mu_i \nabla g_i(\boldsymbol{x}_*) = \boldsymbol{0}, \\ \mu_i g_i(\boldsymbol{x}_*) = 0, \quad i = 1,\, 2,\, \ldots,\, M_2. \end{aligned}$$

Definition 2.2 (**Regularity**). *Consider the minimization problem* (2.9). *A solution* $\boldsymbol{x}_*$ *is called regular if the gradients of the active constraints among the inequality constraints and of the equality constraints are linear independent.*

To illustrate the method of Lagrange multipliers and KKT conditions, we use an example from [3]. Consider

$$\begin{aligned} &\min_{\mathbf{x}} \; x_1 + x_2 \\ &\text{subject to} \;\; x_1^2 + x_2^2 = 1 \end{aligned}$$

For this example, the gradient of the constraint is $\nabla(x_1^2+x_2^2-1) = 2[x_1, x_2]^T$, and the set of gradients of active constrains only consists of $\mathbf{v} = [x_1,\, x_2]^T$. For this set to be linear dependent both x_1 and x_2 must be zero, which is not a solution to the problem. Thus this problem does not have irregular points, an optimal solution exists and the KKT conditions are necessary.

To find the KKT point, we define the Lagrange function

$$L(x_1, x_2, \lambda) = x_1 + x_2 + \lambda(x_1^2 + x_1^2 - 1),$$

where λ is the Lagrange multiplier. The KKT conditions are

$$\begin{aligned} 0 &= \frac{\partial L}{\partial x_1} = 1 + 2\lambda x_1 \\ 0 &= \frac{\partial L}{\partial x_2} = 1 + 2\lambda x_2 \\ 0 &= \frac{\partial L}{\partial \lambda} = x_1^2 + x_2^2 - 1 \end{aligned}$$

From the first two equations we have $x_1 = x_2 = -1/(2\lambda)$. Inserting this into the third equation gives

$$\left(-\frac{1}{2\lambda}\right)^2 + \left(-\frac{1}{2\lambda}\right)^2 = 1,$$

which yield $\lambda = \pm 1/\sqrt{2}$. Thus we obtain the two KKT points, $(1/\sqrt{2}, 1/\sqrt{2})$ and $(-1/\sqrt{2}, -1/\sqrt{2})$. Since $1/\sqrt{2}+1/\sqrt{2} = \sqrt{2}$ and $-1/\sqrt{2}-1/\sqrt{2} = -\sqrt{2}$, it is the latter point that solves the problem. In view of the theorem,

$$\nabla f(\mathbf{x}_*) + \lambda \nabla g(\mathbf{x}_*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} -\frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Chapter 3

Numerical methods

In this chapter, we discuss the methods used for the numerical computations. In the first section we look at descent methods, which seek a point where a function attains its least value. In particular, we look at non-linear conjugate gradient methods. The second section is about finding suitable step sizes, and in the last section we find an expression for the derivatives of the function (2.6) with respect to its parameters.

3.1 Descent methods

We are interested in finding a point $\mathbf{x}_*$ such that an objective function $f : \mathbb{R}^d \to \mathbb{R}$ is minimized (in this and in the next section we write f as the objective function, because it is more natural in the context of optimization (see [25, 22])). To this end, we consider the iteration ([25, 22])

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n, \quad n = 0, 1, 2, \ldots \tag{3.1}$$

The idea here is to move from the current point $\mathbf{x}_n$ to the next point $\mathbf{x}_{n+1}$ by taking a step α_n in the direction of $\mathbf{p}_n$, such that $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$. This

procedure is repeated until for a sufficiently large n, $\mathbf{x}_n$ has converged toward $\mathbf{x}_*$.

The question is to find an appropriate step size and a direction to move towards. We start by discussing the latter problem. If the (search) direction $\mathbf{p}_n$ satisfies

$$\nabla f(\mathbf{x}_n)^T \mathbf{p}_n < 0 \quad \text{for} \quad \nabla f(\mathbf{x}_n) \neq \mathbf{0}, \tag{3.2}$$

$$\mathbf{p}_n = 0 \quad \text{for} \quad \nabla f(\mathbf{x}_n) = \mathbf{0}, \tag{3.3}$$

then it is a descent direction, and (3.1) is a descent method ([25]).

The simplest descent method is the steepest descent method, which corresponds to $\mathbf{p}_n = -\nabla f(\mathbf{x}_n)$. This method always produces a descent direction, and in the next section we will prove that it is convergent under not so strict conditions. However analysis done in [22] indicate that this method can converge slowly.

Some methods that converge faster, are conjugate gradient methods. For these methods the descent direction is recursively given by ([25])

$$\mathbf{p}_{n+1} = -\nabla f(\mathbf{x}_{n+1}) + \beta_{n+1}\, \mathbf{p}_n, \tag{3.4}$$

where β_{n+1} is a scalar and $\mathbf{p}_0 = -\nabla f(\mathbf{x}_0)$. Three well-known formulas for β_{n+1}, due to Fletcher and Reeves (FR), Polak and Ribière (PR) and Dai and Yuan (DY), are

$$\beta_{n+1}^{\mathrm{FR}} = \frac{\|\nabla f(\mathbf{x}_{n+1})\|_2^2}{\|\nabla f(\mathbf{x}_n)\|_2^2}, \tag{3.5}$$

$$\beta_{n+1}^{\mathrm{PR}} = \frac{\nabla f(\mathbf{x}_{n+1})^T (\nabla f(\mathbf{x}_{n+1}) - \nabla f(\mathbf{x}_n))}{\|\nabla f(\mathbf{x}_n)\|_2^2}, \tag{3.6}$$

$$\beta_{n+1}^{\mathrm{DY}} = \frac{\|\nabla f(\mathbf{x}_{n+1})\|_2^2}{(\nabla f(\mathbf{x}_{n+1}) - \nabla f(\mathbf{x}_n))^T \mathbf{p}_n}, \tag{3.7}$$

for $n = 0, 1, \ldots$ The fact that these search directions, with the given formulas for β_{n+1}, are descent directions, depends on the step size, α_n. We want to choose this so that $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$, is not too small and neither too difficult nor too time consuming to compute. In order to calculate the step size, we begin by defining the one-dimensional function ϕ by

$$\phi(\alpha) = f(\mathbf{x}_n + \alpha \mathbf{p}_n), \tag{3.8}$$

where $\mathbf{x}_n$ and $\mathbf{p}_n$ are known. Then the step size can be found by minimizing ϕ along the line $\mathbf{x}_n + \alpha_n \mathbf{p}_n$, and that will solve the problem

$$\text{Find } \alpha = \alpha_n \text{ which minimizes } \phi(\alpha) = f(\mathbf{x}_n + \alpha \mathbf{p}_n). \tag{3.9}$$

Methods that solve this problem are referred to as line search methods ([22]) and this is addressed in section 3.2.

3.2 Line search

In the previous section, we saw that the problem of finding the step size in (3.1) resulted in the one-dimensional minimization problem (3.9). To solve this problem we can either solve the equation $\phi'(\alpha) = 0$ using root-finding methods or use derivative-free minimization methods. Regardless of which method we choose, we will find the exact solution. In this case, we have the following result from [25]:

Theorem 3.1. *If α_n in Eq. (3.1) is the exact solution of the problem (3.9), then*

$$\nabla f(\boldsymbol{x}_{n+1})^T \boldsymbol{p}_n = 0.$$

Proof. Let α_n be as described. Then $\phi'(\alpha_n)$ is necessarily equal to 0. But

we also have, by the chain rule,

$$\begin{aligned}\phi'(\alpha) &= \sum_{i=1}^{N} \frac{\partial}{\partial x_i} f(\mathbf{x}_n + \alpha \mathbf{p}_n) p_n^i \quad p_n^i \text{ is the } i\text{-th entry of } \mathbf{p}_n \\ &= \nabla f(\mathbf{x}_n + \alpha \mathbf{p}_n)^T \mathbf{p}_n\end{aligned}$$

from which the result follows, after inserting α_n and using Eq. (3.1). □

This theorem implies that the gradient methods compute a descent direction for the exact solution of (3.9). Indeed, by taking the inner product between (3.4) and $\nabla f(\mathbf{x}_{n+1})$, we obtain

$$\nabla f(\mathbf{x}_{n+1})^T \mathbf{p}_{n+1} = -\|\nabla f(\mathbf{x}_{n+1})\|_2^2 + \beta_{n+1} \nabla f(\mathbf{x}_{n+1})^T \mathbf{p}_n. \tag{3.10}$$

By the theorem, the second term on the right-hand-side vanishes.

While it is possible to find the exact solution of the problem (3.9), it might be too computational expensive and even unnecessary to do so. Instead, we look to find a step size which satisfies the *Wolfe conditions* ([22]). Such a step size do not solve (3.9) exactly, but still gives a sufficient decrease, and is not too time consuming to compute. The standard Wolfe conditions are

$$f(\mathbf{x}_n + \alpha_n \mathbf{p}_n) \leq f(\mathbf{x}_n) + c_1 \alpha_n \nabla f(\mathbf{x}_n)^T \mathbf{p}_n \tag{3.11}$$

$$\nabla f(\mathbf{x}_n + \alpha_n \mathbf{p}_n)^T \mathbf{p}_n \geq c_2 \nabla f(\mathbf{x}_n)^T \mathbf{p}_n, \tag{3.12}$$

where $0 < c_1 < c_2 < 1$. Figure 3.1 illustrates step sizes satisfying (3.11)-(3.12). The first condition ensures that the function decreases, while the second condition guarantees that the step size is not too small. Among the many methods that compute a step size so that it is not too small and decreases the objective function, is backtracking ([25]). This algorithm starts with an initial candidate for a step size and decreases it by a scale factor ρ

(backtrack step) until (3.11) is satisfied. Algorithm 1 presents a pseudocode of this algorithm.

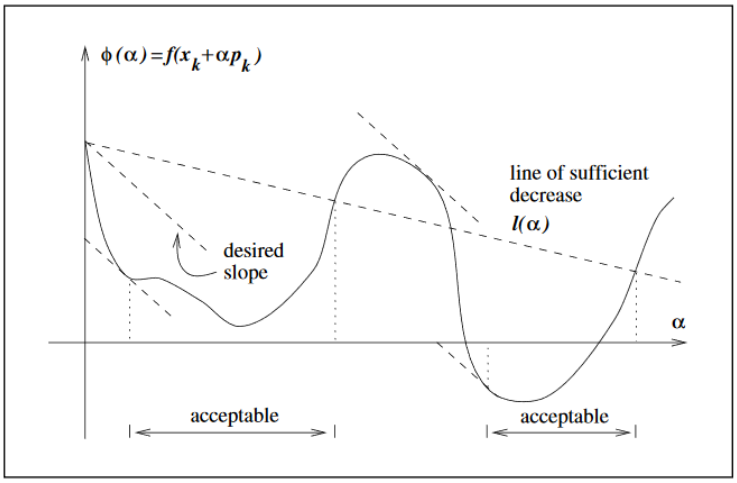


Figure 3.1: Step sizes satifying the standard Wolfe conditions. The line of sufficient deacrease is $l(\alpha) = f(\mathbf{x}_n) + c_1\alpha\nabla f(\mathbf{x}_n)\cdot\mathbf{p}_n$. The picture is taken from [22]

.

Algorithm 1 Backtracking algorithm

Initialize α.

Set ρ and c_1.

while $f(\mathbf{x}_n + \alpha_n\,\mathbf{p}_n) \geq f(\mathbf{x}_n) + c_1\alpha_n\nabla f(\mathbf{x}_n)^T\,\mathbf{p}_n$ **do**

$\alpha = \alpha\rho$

end while

Some gradient methods, like the Dai-Yuan method, compute a descent direction with a step size satisfying the standard Wolfe conditions ([7]). For others, like Fletcher and Reeves, the condition (3.12) must be replaced by the stronger condition

$$|\nabla f(\mathbf{x}_n + \alpha_n\,\mathbf{p}_n)^T\,\mathbf{p}_n| \leq c_2|\nabla f(\mathbf{x}_n)^T\mathbf{p}_n|, \tag{3.13}$$

where now $c_2 < 1/2$, in order to produce a descent direction. The conditions (3.11)-(3.13) are referred to as the strong Wolfe conditions ([22]). The following result (from [22]) verifies that the Fletcher-Reeves' method computes a descent direction.

Lemma 3.1. *If the Fletcher-Reeves method is used with a step size α_n satisfying the strong Wolfe conditions with $0 < c_2 < \frac{1}{2}$, then $\boldsymbol{p}_n$ is a descent direction which satisfies the two inequalities*

$$-\frac{1}{1-c_2} \leq \frac{\nabla f(\boldsymbol{x}_n)^T \boldsymbol{p}_n}{\|\nabla f(\boldsymbol{x}_n)\|_2^2} \leq \frac{2c_2-1}{1-c_2} \tag{3.14}$$

Proof. First, the function $t(x) = (2x-1)/(1-x)$ is monotonically increasing on the interval $[0, 1/2]$, with the properties $t(0) = -1$ and $t(1/2) = 0$. Hence,

$$-1 < \frac{2c_2-1}{1-c_2} < 0.$$

Moreover, $-1 > -1/(1-c_2)$ since c_2 is positive. Thus, the descent condition $\nabla f_n^T \mathbf{p}_n < 0$ (we have denoted f_n instead of $f(\mathbf{x}_n)$ for easier notation) follows immediately once we establish (3.14).

This is done by induction. For $n = 0$, $\mathbf{p}_0 = -\nabla f_0$ and $(\nabla f_0^T \mathbf{p}_0)/\|\nabla f_0\|_2^2 = -1$, so both inequalities are satisfied. Assume that (3.14) is satisfied for some $n > 0$. Then for $n+1$

$$\begin{aligned}
\frac{\nabla f_{n+1}^T \mathbf{p}_{n+1}}{\|\nabla f_{n+1}\|_2^2} &= \frac{\nabla f_{n+1}^T(-\nabla f_{n+1} + \beta_{n+1}^{FR}\mathbf{p}_n)}{\|\nabla f_{n+1}\|_2^2} \\
&= -1 + \beta_{n+1}^{FR}\frac{\nabla f_{n+1}^T \mathbf{p}_n}{\|\nabla f_{n+1}\|_2^2} \\
&= -1 + \frac{\nabla f_{n+1}^T \mathbf{p}_n}{\|\nabla f_n\|_2^2},
\end{aligned}$$

where we have used (3.4) and (3.5). From the second strong Wolfe condition

$$|\nabla f_{n+1}^T \mathbf{p}_n| \leq -c_2 \nabla f_n^T \mathbf{p}_n,$$

so

$$-1 + c_2 \frac{\nabla f_n^T \mathbf{p}_n}{\|\nabla f_n\|_2^2} \leq \frac{\nabla f_{n+1}^T \mathbf{p}_{n+1}}{\|\nabla f_{n+1}\|_2^2} \leq -1 - c_2 \frac{\nabla f_n^T \mathbf{p}_n}{\|\nabla f_n\|_2^2}.$$

From the induction hypothesis, $(\nabla f_n^T \mathbf{p}_n)/\|\nabla f_n\|_2^2 > -1/(1-c_2)$. Inserting this into the above, we get

$$-1 - \frac{c_2}{1-c_2} = \frac{-1}{1-c_2} \leq \frac{\nabla f_{n+1}^T \mathbf{p}_{n+1}}{\|\nabla f_{n+1}\|_2^2} \leq -1 + \frac{c_2}{1-c_2} = \frac{2c_2 - 1}{1-c_2}$$

which shows that (3.14) hold for $n+1$ as well. □

In [11], Gilbert and Nocedal (GN) extended this result to hold for any β_n satisfying $|\beta_n| \leq \beta_n^{\mathrm{FR}}$. Moreover, they show that for such a β_n one has global convergence, and suggested the following modification for the Polak-Ribière method. For $n \geq 1$ let

$$\beta_n^{\mathrm{GN}} = \begin{cases} -\beta_n^{FR} & \text{if } \beta_n^{PR} < -\beta_n^{FR}, \\ \beta_n^{PR} & \text{if } |\beta_n^{PR}| \leq \beta_n^{FR}, \\ \beta_n^{FR} & \text{if } \beta_n^{PR} > \beta_n^{FR}. \end{cases} \tag{3.15}$$

This choice of $\beta_n = \beta_n^{\mathrm{GN}}$ removes the weaknesses that the Fletcher-Reeves and Polak-Ribière methods have separately. Conserning the Fletcher-Reeves method, there is a possibility that the search direction and the gradient can be almost orthogonal. If we additionally have a subsequent small step from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$, the new search direction is not improved compared to the previous one. This also motivates the need for restarts, that we will discussed briefly below. For the Polak-Ribière method, the weakness is that it does not

always produces a descent direction, even with the strong Wolfe conditions. For more details, see [11, 22].

In order to prove global convergence of gradient methods, the next result is the main ingredient.

Theorem 3.2 (Zoutendijk's theorem, [22])**.** *Consider (3.1), with $\boldsymbol{p}_n$ being a descent direction, and α_n being a step size satisfying the standard Wolfe conditions. Suppose f is bounded below in $\mathbb{R}^d$ and is continuous differentiable in an open set $\mathcal{N}$ containing $\mathcal{L} = \{\boldsymbol{x} : f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$, where $\boldsymbol{x}_0$ is the initial guess. Suppose further that ∇f is Lipschitz in $\mathcal{N}$, i.e. there exists a $L > 0$ such that*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2, \quad \textit{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathcal{N}.$$

Then

$$\sum_{n=0}^{\infty} (\cos(\theta_n))^2 \|\nabla f_n\|_2^2 < \infty, \tag{3.16}$$

where θ_n is the angle between $-\nabla f_n$ and $\boldsymbol{p}_n$.

Proof. From the second standard Wolfe condition, (3.1) and the Lipschitz continuity, we have the two inequalities

$$(c_2 - 1)\nabla f_n^T \, \mathbf{P}_n \leq (\nabla f_{n+1} - \nabla f_n)^T \, \mathbf{p}_n \leq \alpha_n L \|\mathbf{p}_n\|_2^2,$$

so

$$\frac{(c_2 - 1)\nabla f_n^T \mathbf{p}_n}{L\|\mathbf{p}_n\|_2^2} \leq \alpha_n.$$

Inserting this into the first standard Wolfe condition yields

$$f_{n+1} \leq f_n - c_1 \frac{(1 - c_2)(\nabla f_n^T \mathbf{p}_n)^2}{L\|\mathbf{p}_n\|_2^2} = f_n - c(\cos(\theta_n))^2 \|\nabla f_n\|_2^2,$$

where the last equality follows from $\|\nabla f_n\|_2\|\mathbf{p}_n\|_2\cos(\theta_n) = -\nabla f_n^T\,\mathbf{p}_n$ and defining $c = c_1(c_2-1)/L$. We continue like this repeatedly for n, $n-1$, ..., and obtain

$$f_0 - f_{n+1} \geq c\sum_{j=0}^{n}(\cos(\theta_j))^2\|\nabla f_j\|_2^2.$$

By assumption, f is bounded below. Thus $f_0 - f_{n+1}$ is less than a positive constant, for all n. Hence, letting $n \to \infty$ in the above sum gives

$$\sum_{j=0}^{\infty}(\cos(\theta_j))^2\|\nabla f_j\|_2^2 < \infty$$

□

A consequence of this theorem is that for a large n we have, by the n-th term test ([1]),

$$(\cos(\theta_n))^2\|\nabla f_n\|_2^2 = 0.$$

This, in turn, means that either $\cos(\theta_n) = 0$ or $\|\nabla f_n\|_2 = 0$. For the former, this implies that ∇f_n and $\mathbf{p}_n$ are orthogonal. For the steepest descent method, $\mathbf{p}_n$ and ∇f_n are always parallel. Hence it converges towards a point for which $\|\nabla f_n\|_2 = 0$. Furthermore we have:

Theorem 3.3 (Convergence of gradient methods with $|\beta_n| \leq \beta_n^{\mathrm{FR}}$, [11])**.** *Consider a gradient method with $|\beta_n| \leq \beta_n^{FR}$ and a step size satisfying the strong Wolfe conditions. Assume f is Lipschitz in $\mathcal{N}$ and that $\mathcal{L}$ is bounded, where $\mathcal{L}$ and $\mathcal{N}$ are from Zoutendijk's theorem. Then*

$$\liminf_{n\to\infty}\|\nabla f_n\|_2 = 0 \tag{3.17}$$

Proof. Assume the contrary, that is, (3.17) does not hold. Then there exists a $\gamma > 0$ such that

$$\|\nabla f_n\|_2 \geq \gamma, \tag{3.18}$$

for all large n. Multiply the right inequality of (3.14) by $-\|\nabla f_n\|_2/\|\mathbf{p}_n\|_2$ to obtain

$$\cos(\theta_n) \geq \frac{1 - 2c_2 \|\nabla f_n\|_2}{(1 - c_2) \|\mathbf{p}_n\|_2}.$$

Then, by Zoutendijk's theorem,

$$\sum_{n=0}^{\infty} \frac{\|\nabla f_n\|_2^4}{\|\mathbf{p}_n\|_2^2} < \infty.$$

By the second strong Wolf condition and (3.14), we have

$$|\nabla f_n^T \mathbf{p}_{n-1}| \leq -c_2 \nabla f_{n-1}^T \mathbf{p}_{n-1} \leq \frac{c_2}{1 - c_2} \|\nabla f_{n-1}\|_2.$$

Thus, by (3.4) and that $|\beta_n| \leq \beta_n^{\mathrm{FR}}$,

$$\begin{aligned}
\|\mathbf{p}_n\|_2^2 &\leq \|\nabla f_n\|_2^2 + 2|\beta_n| \, |\nabla f_n^T \mathbf{p}_{n-1}| + (\beta_n)^2 \|\mathbf{p}_{n-1}\|_2^2 \\
&= \|\nabla f_n\|_2^2 + \frac{2c_2}{1 - c_2} |\beta_n| \, \|\nabla f_{n-1}\|_2^2 + (\beta_n)^2 \|\mathbf{p}_{n-1}\|_2^2 \\
&\leq \left(\frac{1 + c_2}{1 - c_2} \right) \|\nabla f_n\|_2^2 + (\beta_n)^2 \|\mathbf{p}_{n-1}\|_2^2.
\end{aligned}$$

Denoting $c = (1 + c_2)/(1 - c_2)$ and applying this relation repeatedly, we obtain

$$\begin{aligned}
\|\mathbf{p}_n\|_2^2 &\leq c\|\nabla f_n\|_2^2 + (\beta_n)^2 (c\|\nabla f_{n-1}\|_2^2 + (\beta_{n-1})^2 \|\mathbf{p}_{n-2}\|_2^2) \\
&= c\big(\|\nabla f_n\|_2^2 + (\beta_n)^2 \|\nabla f_{n-1}\|_2^2\big) + (\beta_n)^2 (\beta_{n-1})^2 \|\mathbf{p}_{n-2}\|_2^2 \\
&\leq c\Big(\|\nabla f_n\|_2^2 + (\beta_n)^2 \big(\|\nabla f_{n-1}\|_2^2 + (\beta_{n-1})^2 \|\nabla f_{n-2}\|_2^2\big) \Big) \\
&\quad + (\beta_n)^2 (\beta_{n-1})^2 (\beta_{n-2})^2 \|\mathbf{p}_{n-3}\|_2^2 \\
&\leq \dots \\
&\leq c\|\nabla f_n\|_2^4 \sum_{j=0}^{n} \frac{1}{\|\nabla f_j\|_2^2}.
\end{aligned}$$

The last inequality follows from $|\beta_n| < \beta_n^{\mathrm{FR}}$.

Since f is Lipschitz, we have by the mean value theorem that there is a $M > 0$ such that $\|\nabla f(\mathbf{x})\|_2 < M$ for all $\mathbf{x} \in \mathcal{L}$. Together with the assumption that the method has not converged, we thus obtain the bound

$$\|\mathbf{p}_n\|_2^2 \leq \frac{cM}{\gamma} n,$$

and hence

$$\sum_{n=1}^{\infty} \frac{1}{\|\mathbf{p}_n\|_2^2} \geq \frac{\gamma}{cM} \sum_{n=1}^{\infty} \frac{1}{n}.$$

However, this implies that

$$\sum_{n=1}^{\infty} \frac{1}{n} \leq \sum_{n=1}^{\infty} \frac{1}{\|\mathbf{p}_n\|_2^2} < \infty,$$

which is not true. Therefore (3.18) does not hold, which means that (3.17) is verified. □

Using the same strategy as in the proof above, one can show that the Dai-Yuan method also converges towards a point for which

$$\liminf_{n \to \infty} \|\nabla f_n\|_2 = 0,$$

but the step size is now required to satisfy the standard Wolfe conditions. For the proof, see [7].

A weakness with these results is that they only guarantee convergence towards a critical point, but this does not guarantee that the the critical point actually minimizes the objective function. Moreover, if we have reached a minimum, there are no indicators that can explain whether this is global or local.

The rate of convergence for non-linear conjugate gradient methods is linear, but by restarting periodically at every k-th step (i.e. setting $\beta_k = 0$ in (3.4)), the rate of convergence becomes k-step quadratic convergence, $\|\mathbf{x}_{k+n} - \mathbf{x}_*\|_2 = \mathcal{O}(\|\mathbf{x}_n - \mathbf{x}_*\|_2^2)$ ([22]). However, in computational practice, strategies for restarts are based on other considerations than iteration count ([22]).

3.3 Adjoint method

The last thing to discuss is how to compute of the derivatives of J, with respect to its parameters. For this we use the adjoint method, introduced in the 1970s to efficently compute the gradient of functions with respect to its parameters ([24]).

To derive the adjoint problem, we recall Lagrange function (2.8)

$$L(u, \mathbf{d}, \lambda) = \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\iint_{\Omega}\frac{(u(\mathbf{x},t;\mathbf{d}) - u_m^*(t)\,)^2\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}d\mathbf{x}dt + \int_{t=0}^{t=T}\iint_{\Omega}\lambda\Big(u_t - D\Delta u + \mathbf{w}\cdot\nabla u - F\Big)d\mathbf{x}dt.$$

Using integration by parts we get

$$\begin{aligned}
\int_{t=0}^{t=T}\iint_{\Omega}\lambda u_t d\mathbf{x}dt &= \iint_{\Omega}\big(\lambda u|_{t=0}^{t=T} - \int_{t=0}^{t=T} u\lambda_t dt\big)d\mathbf{x},\\
\int_{t=0}^{t=T}\iint_{\Omega}\lambda\ (\mathbf{w}\cdot\nabla u)\ d\mathbf{x}dt &= \int_{t=0}^{t=T}\int_{\partial\Omega} u\lambda\,(\mathbf{w}\cdot\mathbf{n})ds - \iint_{\Omega} u\,(\nabla\cdot(\lambda\mathbf{w}))d\mathbf{x}\,dt\\
&= -\int_{t=0}^{t=T}\iint_{\Omega} u\ (\nabla\cdot(\lambda\mathbf{w}))d\mathbf{x}dt\\
&= -\int_{t=0}^{t=T}\iint_{\Omega} u\ (\mathbf{w}\cdot\nabla\lambda)\,d\mathbf{x}dt,
\end{aligned}$$

after using the boundary condition (2.3), the identity (B.1), and that the

velocity field is divergence free. By Green's second identity (see Appendix B), together with the boundary condition (2.3), we have

$$\iint_\Omega \lambda D\Delta u d\mathbf{x} = \iint_\Omega uD\Delta\lambda d\mathbf{x} + \int_{\partial\Omega} D\lambda\ (\mathbf{n}\cdot\nabla u)ds.$$

where $\mathbf{n}\cdot\nabla u$ is the directional derivative in the outward normal direction and ds is the element of arc length. Then

$$\begin{aligned} L = \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\iint_\Omega \frac{(u(\mathbf{x},t)-u_m^*(t)\,)^2\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}d\mathbf{x}dt \\ - \int_{t=0}^{t=T}\iint_\Omega u\big(\lambda_t + \mathbf{w}\cdot\nabla\lambda + D\Delta\lambda\big)d\mathbf{x}dt \\ - \int_{t=0}^{t=T}\iint_\Omega \lambda F d\mathbf{x}dt \\ + \iint_\Omega \lambda u|_{t=0}^{t=T} d\mathbf{x} \\ + \int_{\partial\Omega}\lambda\ (\mathbf{n}\cdot\nabla u)ds. \end{aligned}$$

By writing $(u-u_m^*)^2 = u^2 - 2\,u\,u_m^* + (u_m^*)^2 = u^2 - 2\,u\,(u_m^*) + (u_m^*)^2 + u^2 - u^2 = 2\,u\,(u-u_m^*) + (u_m^*)^2 - u^2$, we get

$$\begin{aligned} L = \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\iint_\Omega \frac{\Big((u_m^*(t))^2 - (u(\mathbf{x},t))^2\Big)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}d\mathbf{x}dt \\ + \int_{t=0}^{t=T}\iint_\Omega u\bigg(-\lambda_t - \mathbf{w}\cdot\nabla\lambda - D\Delta\lambda + \sum_{m=1}^{N_m}\frac{(u-u_m^*)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}\bigg)d\mathbf{x}dt \\ - \int_{t=0}^{t=T}\iint_\Omega \lambda F d\mathbf{x}dt \\ + \iint_\Omega \lambda u|_{t=0}^{t=T} d\mathbf{x} \\ + \int_{\partial\Omega}\lambda(\mathbf{n}\cdot\nabla u)ds. \end{aligned}$$

We choose λ to be the solution to the IBVP

$$-\lambda_t - \mathbf{w}\cdot\nabla\lambda - D\Delta\lambda + \sum_{m=1}^{N_m}\frac{(u-u_m^*)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2} = 0, \tag{3.19}$$

$$\lambda(\mathbf{x},T) = 0, \tag{3.20}$$

$$\lambda|_{\partial\Omega} = 0. \tag{3.21}$$

Thus the Lagrangian simplifies to

$$\begin{aligned} L &= \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\iint_\Omega \frac{\Big((u_m^*(t))^2 - (u(\mathbf{x},t))^2\Big)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}d\mathbf{x}dt \\ &\quad - \int_{t=0}^{t=T}\iint_\Omega \lambda F d\mathbf{x}dt \qquad (3.22) \\ &= \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\iint_\Omega \frac{\Big((u_m^*(t))^2 - (u(\mathbf{x},t))^2\Big)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}d\mathbf{x}dt \\ &\quad - \int_{t=0}^{t=T}\iint_\Omega \lambda\sum_{i=1}^{N_s} a_i\delta(\mathbf{x}-\boldsymbol{\xi}_i)d\mathbf{x}dt \qquad (3.23) \\ &= \frac{1}{2}\sum_{m=1}^{N_m}\int_{t=0}^{t=T}\frac{(u(\mathbf{x}_m,t))^2 - (u_m^*(t)\,)^2}{w_m^2}dt \\ &\quad - \int_{t=0}^{t=T}\sum_{i=1}^{N_s} a_i\lambda(\boldsymbol{\xi}_i,t)dt. \qquad (3.24) \end{aligned}$$

The IBVP (3.19)-(3.21) is the adjoint problem. Note that this problem exists for $t\in[T,0]$. We would like to solve it for $[0,T]$. Thus we introduce

$\tau = T - t$, and get the IBVP

$$\tilde{\lambda}_\tau - \tilde{\mathbf{w}} \cdot \nabla\tilde{\lambda} - D\Delta\tilde{\lambda} = -\sum_{m=1}^{N_m} \frac{(\tilde{u} - \tilde{u}_m^*)\delta(\mathbf{x} - \mathbf{x}_m)}{w_m^2}, \tag{3.25}$$

$$\tilde{\lambda}|_{\tau=0} = 0, \tag{3.26}$$

$$\tilde{\lambda}|_{\partial\Omega} = 0, \tag{3.27}$$

where $\tilde{\lambda}(\tau) = \lambda(T - t)$ etc.

Owing to $\nabla_{\mathbf{d}} L = \nabla_{\mathbf{d}} J$ when the parameters are correct, we find that the derivatives of the objective function with respect to the parameters are

$$\frac{\partial J}{\partial a_i} = -\int_{t=0}^{t=T} \tilde{\lambda}(\boldsymbol{\xi}_i, t)dt,\ i = 1,\ ...,\ N_s \tag{3.28}$$

$$\frac{\partial J}{\partial \boldsymbol{\xi}_i} = -\int_{t=0}^{t=T} a_i \nabla\tilde{\lambda}(\boldsymbol{\xi}_i, t)dt,\ i = 1,\ ...,\ N_s. \tag{3.29}$$

The computations above are based on the source function (2.4). If we had used (2.5) instead, then

$$\lambda F = \lambda \sum_{i=1}^{N_s} a_i \Phi(\|\mathbf{x} - \boldsymbol{\xi}_i\|_2; \epsilon_i).$$

Hence the terms in the last sum in (3.23) are replaced by

$$\begin{aligned} \iint_\Omega \lambda a_i \Phi(\|\mathbf{x} - \boldsymbol{\xi}_i\|_2; \epsilon_i)d\mathbf{x} &\approx a_i \iint_{\|\mathbf{x}-\boldsymbol{\xi}_i\|_2<\epsilon_i} \lambda(\boldsymbol{\xi}_i)d\mathbf{x} \\ &= a_i\lambda(\boldsymbol{\xi}_i) \iint_{\|\mathbf{x}-\boldsymbol{\xi}_i\|_2<\epsilon_i} d\mathbf{x}, \end{aligned} \tag{3.30}$$

for $i = 1, 2, \ldots, N_s$. For the source function (2.5), the intensity is averaged over the neighbourhood of the source location. That is,

$$a_i = \hat{a}_i \Big/ \iint_{\|\mathbf{x}-\boldsymbol{\xi}_i\|_2<\epsilon_i} d\mathbf{x}, \tag{3.31}$$

see [26]. Here, $\hat{a}_i$ represents the source intensity and a is the averaged source intensity. Inserting this into (3.30) yields

$$\left(\hat{a}_i \Big/ \iint_{\|\mathbf{x}-\boldsymbol{\xi}_i\|_2<\epsilon_i} d\mathbf{x}\right)\lambda(\boldsymbol{\xi}_i)\iint_{\|\mathbf{x}-\boldsymbol{\xi}_i\|_2<\epsilon_i} d\mathbf{x} = \hat{a}_i\lambda(\boldsymbol{\xi}_i).$$

It now follows that for the averaged intensity (3.31), it is the numerator, $\hat{a}_i$, we seek. Moreover, the derivatives of J with respect to its parameters are given by (3.28) and (3.29), but with a_i replaced by $\hat{a}_i$.

By the KKT conditions, we must find the sources locations and intensities such that

$$\begin{aligned}
0 &= \frac{\partial L}{\partial \lambda} = \int_{t=0}^{t=T}\iint_{\Omega}\left(\frac{\partial u}{\partial t} - D\Delta u + \mathbf{w}\cdot\nabla u - F\right)d\mathbf{x}dt \\
0 &= \frac{\partial L}{\partial u} = \int_{t=0}^{t=T}\iint_{\Omega}\left(\tilde{\lambda}_\tau - \tilde{\mathbf{w}}\cdot\nabla\tilde{\lambda} - D\Delta\tilde{\lambda} + \sum_{m=1}^{N_m}\frac{(\tilde{u}-\tilde{u}_m^*)\delta(\mathbf{x}-\mathbf{x}_m)}{w_m^2}\right)d\mathbf{x}dt \\
0 &= \frac{\partial L}{\partial a_i} = -\int_{t=0}^{t=T}\lambda(\boldsymbol{\xi}_i,t)dt,\ i=1,...,N_s \\
0 &= \frac{\partial L}{\partial \boldsymbol{\xi}_i} = -\int_{t=0}^{t=T} a_i\nabla\lambda(\boldsymbol{\xi}_i,t)dt,\ i=1,\ ...\ ,N_s
\end{aligned}$$

is satisfied.

We have now everything ready for estimating the source locations and intensities. Algorithm 2 states all the steps.

Algorithm 2 Gradient method to estimate source location and intensity

Make an initial guess of F

Solve the IBVP (2.1)-(2.3), based on the guess

Solve the adjoint IBVP (3.25)- (3.27)

Initialize $\mathbf{p}_0 = -\nabla J(\mathbf{x}_0)$

for $n = 0, 1, 2, \ldots$ **do**

Compute α_n

$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n$

Solve the IBVP (2.1)-(2.3)

Solve the adjoint IBVP (3.25)- (3.27)

Compute $\nabla J(\mathbf{x}_n)$

Compute β_{n+1}

$\mathbf{p}_{n+1} = -\nabla J(\mathbf{x}_{n+1}) + \beta_{n+1}\mathbf{p}_n$

end for

In this algorithm, $\mathbf{x}_n$ contains all the unknown parameters at step n and

$$\nabla J(\mathbf{x}_n) = \begin{bmatrix} -\int_0^T \lambda(\boldsymbol{\xi}_1, t)dt \\ -\int_0^T a_0 \nabla_x \lambda(\boldsymbol{\xi}_1, t)dt \\ -\int_0^T a_0 \nabla_y \lambda(\boldsymbol{\xi}_1, t)dt \\ \vdots \\ -\int_0^T \lambda(\boldsymbol{\xi}_{N_m}, t)dt \\ -\int_0^T a_{N_m} \nabla_x \lambda(\boldsymbol{\xi}_{N_m}, t)dt \\ -\int_0^T a_{N_m} \nabla_y \lambda(\boldsymbol{\xi}_{N_m}, t)dt \end{bmatrix}.$$

$\nabla_x \lambda$ is the first component of the gradient of λ, and similarly, $\nabla_y \lambda$ is the second component.

Chapter 4

Velocity Field in one direction

In this chapter we illustrate algorithm 2 on a few simple cases.

For the first three sections, we consider the simulated concentration plotted in figure 4.1. This could be, for example, a pollutant in water.

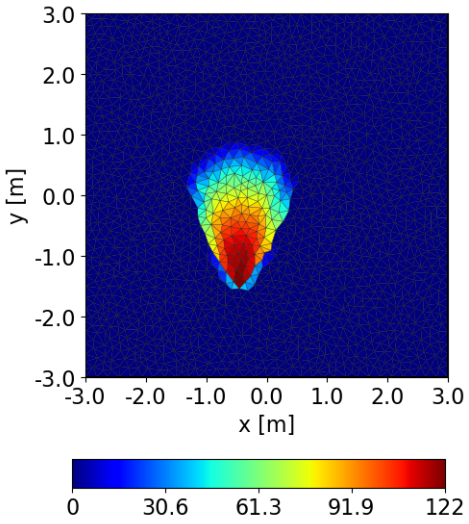


Figure 4.1: A snapshot of a concentration at $T = 8$.

Ω is a 6×6 area, discretized with triangles. The largest area of the triangles is 0.017 $[m^2]$. The true source function is

$$F = 2\delta(\mathbf{x} - [-0.5, -1.45]^T)$$

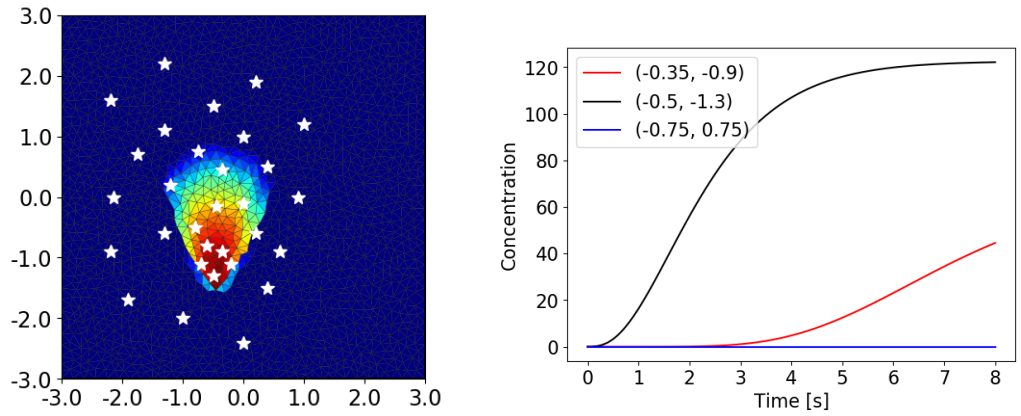
The time interval is $[0,\, 8]\,[s]$, with a time step of $dt = 0.1$. The diffusion coefficient is set to 10^{-9} $[m^2/s]$ (a value it has in water ([26])). The velocity field is $\mathbf{w} = \nabla\phi$, where ϕ is a velocity potential ([17]), given by the Poisson problem

$$\begin{aligned} \Delta\phi &= 0, \\ \phi(\text{bottom}) = 0.5,\ \phi(\text{top}) &= 0, \\ \mathbf{n}_{\text{left}} \cdot \nabla\phi(\text{left}) &= 0, \\ \mathbf{n}_{\text{right}} \cdot \nabla\phi(\text{right}) &= 0. \end{aligned} \tag{4.1}$$

Figure 4.3a displays the velocity field.

We measure at $N_m = 30$ different locations, marked with the white stars in figure 4.2a. The coordinates are given in Appendix A. Figure 4.2b depicts the temporal evolution of the concentration at three different measurement locations.

The weights (see figure 4.3b) are chosen as $\mathrm{w}_m = \mathrm{std}(u_m^*) + 3$, where $\mathrm{std}(u_m^*)$ denotes the standard deviation of u_m^*. For backtracking, α_n has been initialized to be 1, with $c_1 = 10^{-5}$ and $\rho = 0.5$. Additionally with backtracking, it may happen for some candidate step size, that the next point $\mathbf{x}_n + \alpha_n \mathbf{p}_n$ is outside of Ω. In such a case, motivated by the fact that the scale factor is not required to be the same at every step ([22]), the step size is scaled by 0.02.

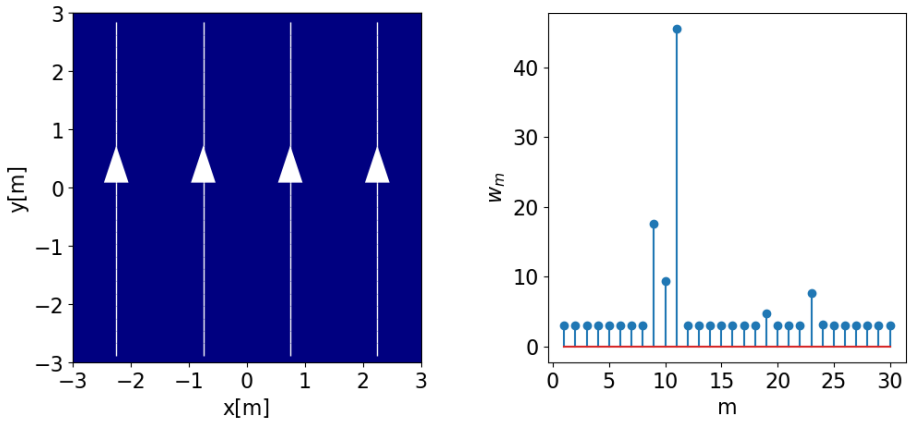


(a) (b)

Figure 4.2: In (a), the measurement locations are plotted as white stars in Ω. In plot (b), we see the temporal evolution of the concentration at three different measurement locations.

To solve the IBVP we use the upwind scheme, with a Crank-Nicolson scheme in time, in the Python package pyGIMLI ([26]). Simpson's rule ([27]) is used to evaluate the integrals. The gradient methods we use are GN, DY (these are listed as recommended methods to use in [22]) and the steepest descent method (SD).

We restart whenever $\mathbf{p}_n^T \nabla J \geq 0$ (this criterion is taken from [2]), or if the first Wolfe condition (3.11) has not been satisfied after 15 iterations in backtracking. We also use Tikhonov regularization, on the form $\tau \|\mathbf{x}_n\|_2^2$, where the coefficient $\tau = 0.0005$ is the regularization parameter. With such a regularization, the gradient of J becomes $\nabla J + 2\tau \mathbf{x}_n$. Finally, we stop if we have not converged within 50 iterations or if $\|\nabla J + 2\tau \mathbf{x}_n\|_2$ is less than 0.005.



(a) (b)

Figure 4.3: In (a), we have plotted the velocity field, obtained by solving the Poisson problem (4.1). In (b) we have plotted of the weights used for the computations.

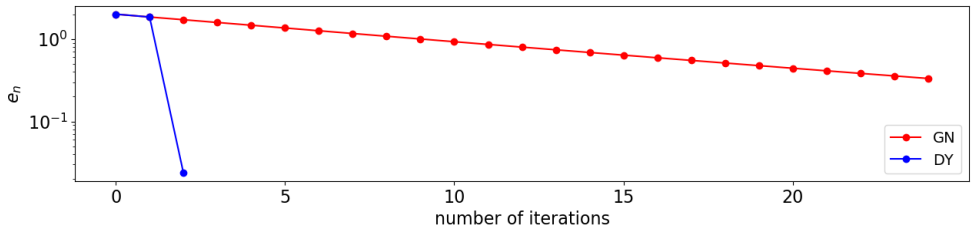
4.1 Fixed position

We begin by identifying the intensity, given that the position is known. This is actually a linear problem, and can be solved more efficiently by a linear optimization method rather than by a non-linear method. Nevertheless, this is a test before we attempt to find both intensity and location of the source.

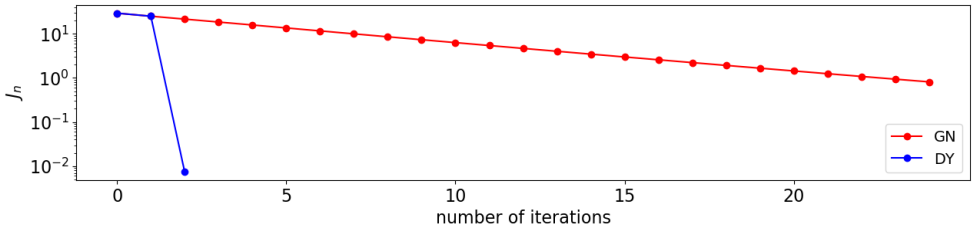
Using 0 as the initial guess, table 4.1 reports estimated values of the intensity for GN and DY. Figure 4.4 visualizes the error of the intensity, defined as $|a_n - a_{\text{true}}|$, the objective function and the 2-norm of the gradient at every iteration on log scale. With 10 as the initial guess, table 4.2 and figure 4.5 present the same information. For both initial guesses, the values for SD are almost identical to GN.

Table 4.1: True value and estimated values of intensity, using GN and DY. The initial guess is 0.

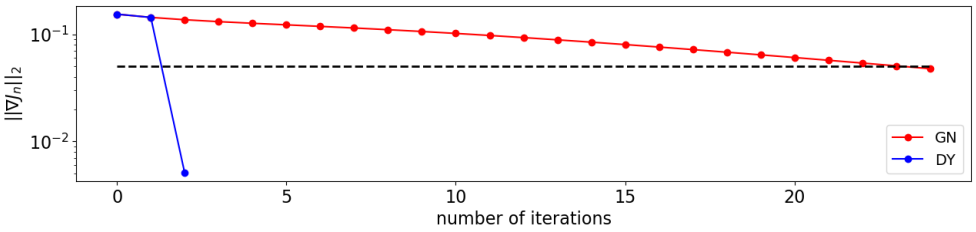
	True	GN	DY
a	2	1.6679	2.0237



(a)



(b)

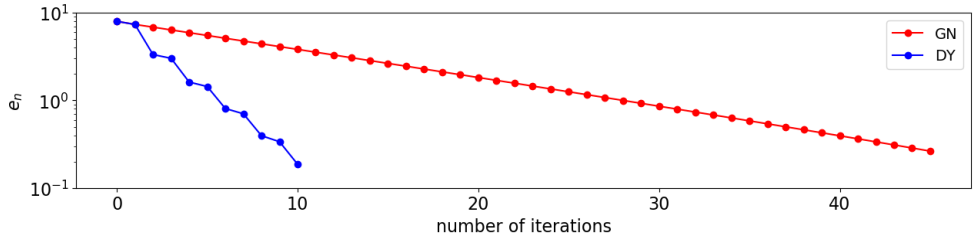


(c)

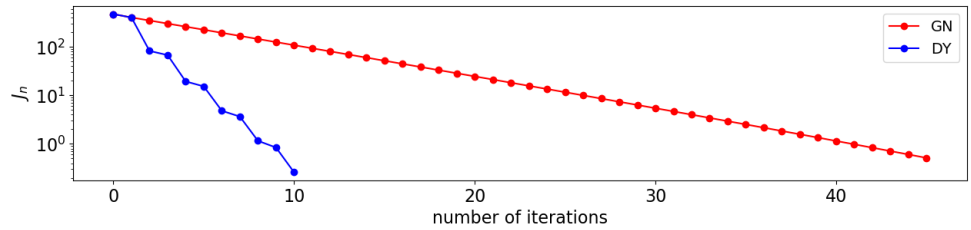
Figure 4.4: Plot of the error $|a_n\text{-}a_{true}|$ for intensity (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)) on log scale at every iteration. The position is known. The dotted line in the gradient plot is the stopping tolerance. The initial guess is 0.

Table 4.2: True value and estimated values of intensity, using GN and DY. The initial guess is 10.

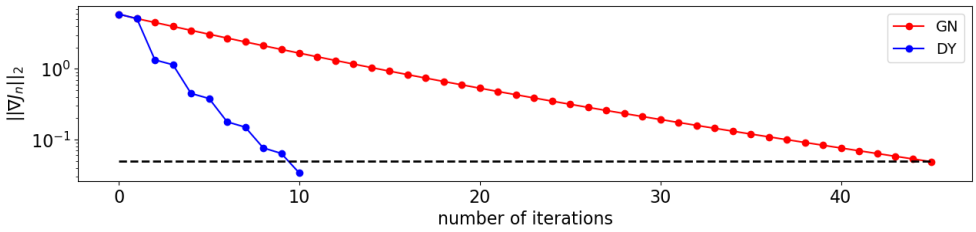
	True	GN	DY
a	2	2.2633	2.1864



(a)



(b)



(c)

Figure 4.5: Plot of the error $|a_n\text{-}a_{true}|$ for intensity (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)) on log scale at every iteration. The position is known. The dotted line in the gradient plot is the stopping tolerance. The initial guess is for intensity was 10.

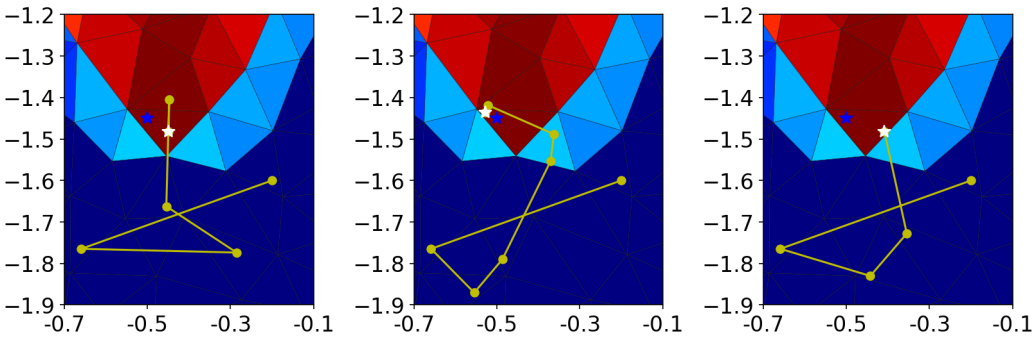
We can notice the difference between the two methods. With 0 as the initial guess, DY uses two steps to reach an intensity of 2.02. GN, on the other hand, has reached after 24 iterations, an intensity with an error approximately at 0.4. Using 10 as the initial guess, GN needs almost 50 iterations to get the 2-norm of the gradient less than the tolerance, whereas DY only uses 10 iterations.

The explanation to this is that the gradients are close to each other at every step. In this situation, GN will always compute a small β_n^{GN}. By formula (3.4), the new search direction, $\mathbf{p}_{n+1}$, is then essentially the negative gradient. Since the gradient has a small first entry, we thus have a slow convergence. For this reason, we do not use GN and SD for the two other intensity cases (in the case of known position) in this thesis.

DY handles this differently. At the first two approximations, since the two gradients are close, $(\nabla J_1 - \nabla J_0)^{-1}$ is large. This, in turn, implies that $\mathbf{p}_1$ also is large. Consequently, we have a large next step. However, by the formula (3.7), the following step will then be small. This behaviour will repeat itself, and leads to the "stair-like" pattern, as illustrated in figure 4.5.

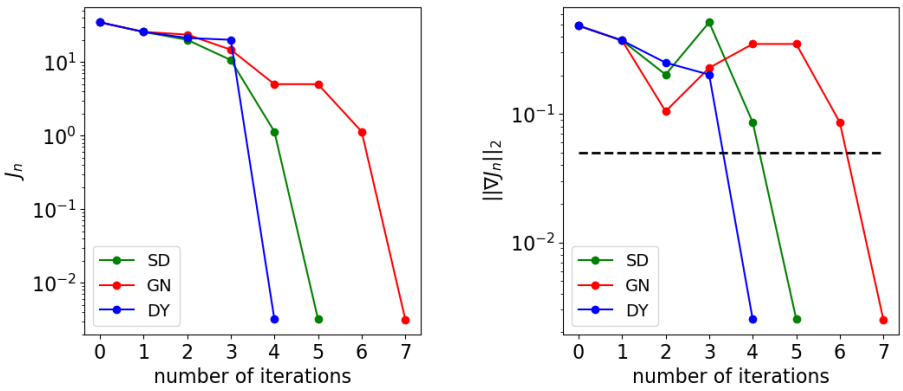
4.2 Fixed intensity

In this section we assume that the intensity is known, making the goal to find the source position. Figure 4.6 displays the iterations in the domain for all the three methods. The initial guess is $(x, y) = (-0.2, -1.6)$. The white star is where the particular method ends up, and the blue star is the true source position. All three methods converge to the correct triangle. Figure 4.7 shows the corresponding objective function and the 2-norm of the gradient at each step. Table 4.3 presents the estimated location.



(a) (b) (c)

Figure 4.6: Iterations in the domain for SD (plot (a)), GN (plot (b)) and DY (plot (c)). The white star is where the particular method ends up, and the blue star is the correct source position. The initial guess is $(-0.2, -1.6)$.



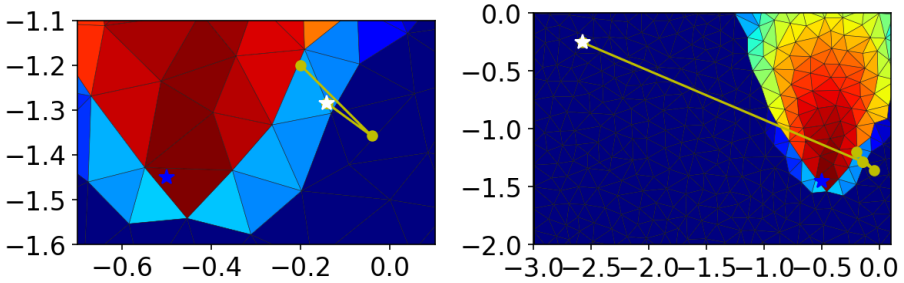
(a) (b)

Figure 4.7: Plot of the objective function (plot (a)) and of the 2-norm of the gradient (plot (b)) on log scale at every iteration. The intensity is known. The dotted line in the gradient plot is the tolerance. The initial guess is $(-0.5, -1.45)$.

Table 4.3: True value and estimated values of source location, using SD, GN and DY. The initial guess is $(-0.2, -1.6)$.

	True	SD	GN	DY
(x, y)	(-0.5, -1.45)	(-0.4489, -1.4825)	(-0.5275, -1.4354)	(-0.4084, -1.4823)

Starting from $(-0.2, -1.2)$, figure 4.8 shows two situations that will cause problems when attempting to find the source location. In figure 4.8a, GN got stuck at $(-0.14, -1.29)$. The same happened for SD. It is not clear why it happens. Possible reasons could be that the time step should be smaller, that the triangles should be smaller or that the simulation should have lasted longer. Another reason could be that the weight is not appropriately chosen, and ends up being too large or too small. In figure 4.8b, DY ended up at $(-2.49, -0.28)$ with an almost vanishing gradient. The problem here is that we obtained a search direction with large entries together with a large step size.



(a) (b)

Figure 4.8: Two problems finding the source location. In plot (a) GN got stuck, and in plot (b) DY got pushed far away.

These problems are unavoidable. Nevertheless, there are some possible solutions that could partially solve them. Regarding the first problem where we got stuck, we could resolve this by taking a step in a random direction and restart. A solution for the second problem could be to choose the initial step size to be large if the entries in $\mathbf{p}_n$ are small, and conversely small if the entries are large. If so, we will not take too large steps. Another solution is to use a suitable regularization parameter. This could help the gradient not to vanish at the wrong locations.

4.3 Position and intensity

In this section, we look to find both position and intensity of the source. As the initial guess, we start at $(-0.3, -1.25)$, with an intensity of 2.3. The estimated values of source location and intensity are presented in table 4.4. Figure 4.9 exhibits the iterations in the domain for the three methods, and figure 4.10 displays the error of intensity, objective function and the 2-norm of the gradient. The outcome for SD is another example of what we discussed for DY in the previous section.

Table 4.4: True value and estimated values of intensity and source location, using SD, GN and DY. The initial guess is 2.3 for intensity and $(-0.3, -1.25)$ for position.

	True	SD	GN	DY
a	2	2.1711	2.1591	2.1306
(x,y)	(-0.5, -1.45)	(-1.1872, -1.5894)	(-0.4779 , -1.4783)	(-0.4468, -1.4795)

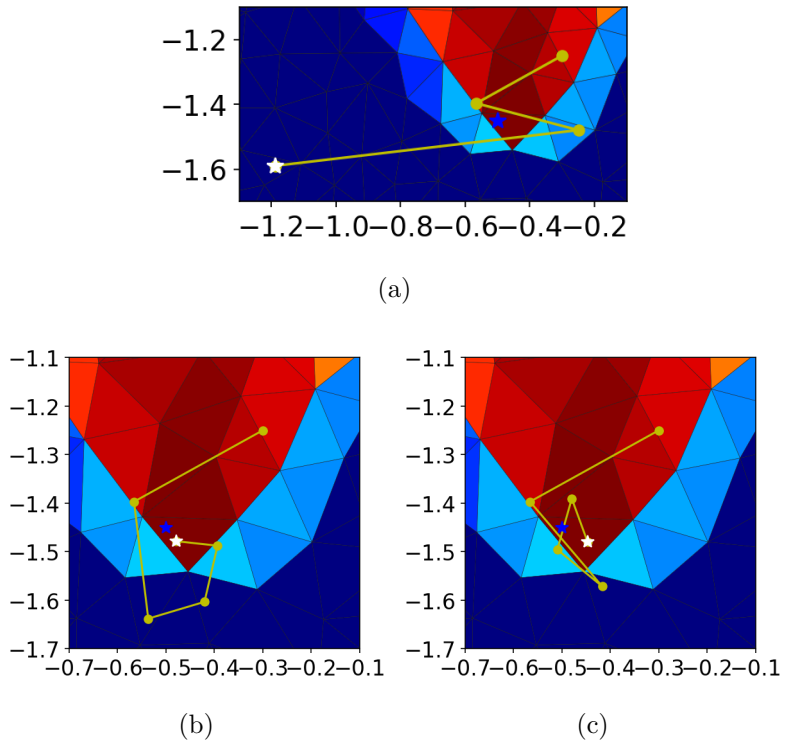
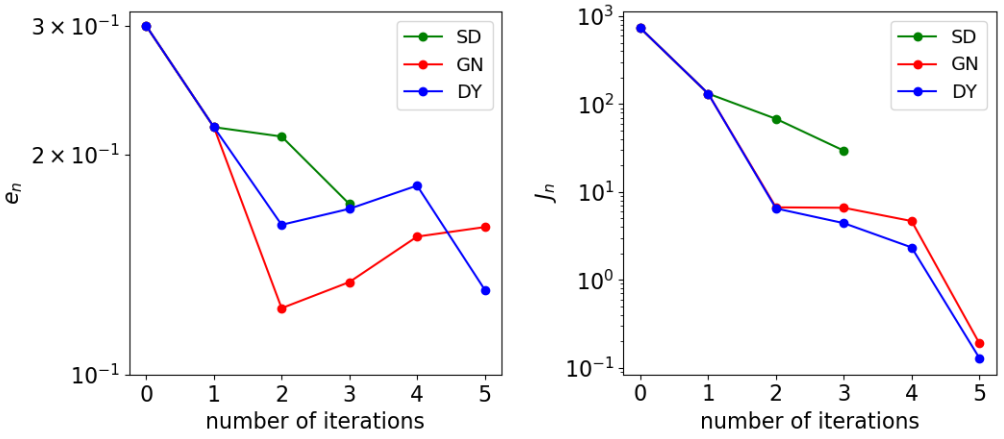
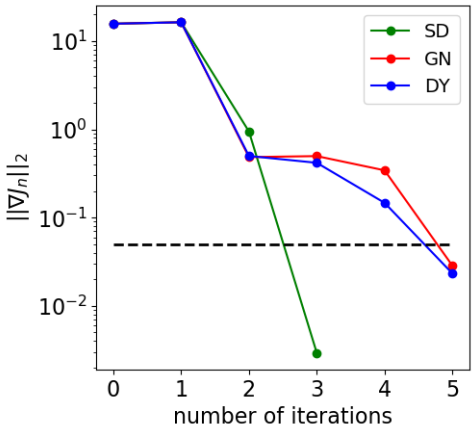


Figure 4.9: Iterations in the domain for SD (plot (a)), GN (plot (b)), and DY (plot (c)). Both position and intensity are unknown. The white star is located where the particular method ends up, and the blue star is the true source position. The initial guess is 2.3 for intensity and $(-0.3, -1.25)$ for position.



(a)

(b)



(c)

Figure 4.10: Plot of the error $|a_n\text{-}a_{true}|$ for intensity (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)) on log scale at every iteration. Both position and intensity are unknown. The dotted line in the gradient plot is the stopping tolerance. The initial guess is 2.3 for intensity and $(-0.3, -1.25)$ for position.

In real-life problems, measurements always contain some noise ([2]). By adding white (Gaussian) noise ([4]) with a standard deviation of 2 to the measurements, the results that we obtained in the various cases are almost identical.

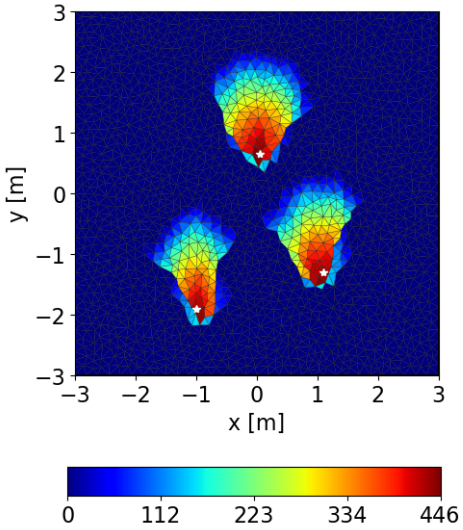


Figure 4.11: A snapshot of three simulated concentrations at $T = 4.5$. The white stars correspond to the locations of the measurements.

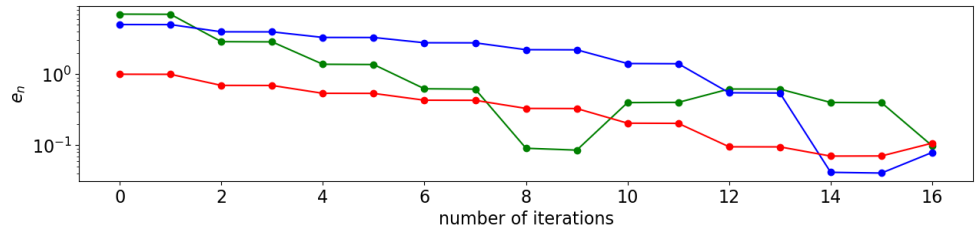
4.4 Several intensities

In this last section of this chapter we look to find several intensities simultaneously, knowing the true source positions. In figure 4.11 we have plotted three concentrations, together with three measurement locations. See Appendix A for the (x, y)-coordinates. The true source function is

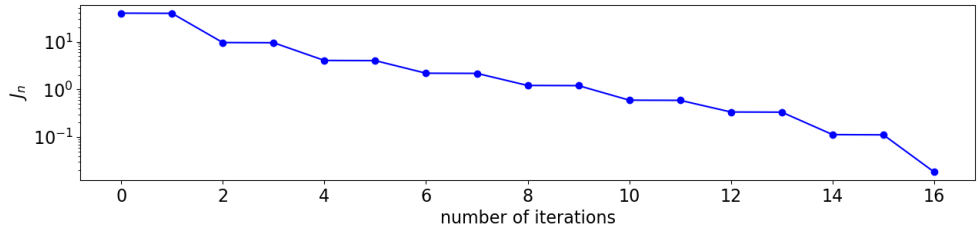
$$F = 3\delta(\mathbf{x} - [-1, -2]^T) + 5\delta(\mathbf{x} - [1, -1.5]^T) + 6\delta(\mathbf{x} - [0, 0.5]^T)$$

In this section, we simulated to T=4.5 with dt=0.2, and considered (2.6) with w_m=std(u_m^*)+1. The regularization parameter was chosen as τ=0.00005, and we stopped when $\|\nabla J + 2\tau\mathbf{x}_n\|_2 < 0.01$. The diffusion coefficient is 10^{-9} and the velocity field is again the gradient of the velocity potential, which we obtained by solving (4.1). Using $10, 0$ and 5 as initial guesses, the errors, the objective function and the 2-norm of the gradient are plotted in figure

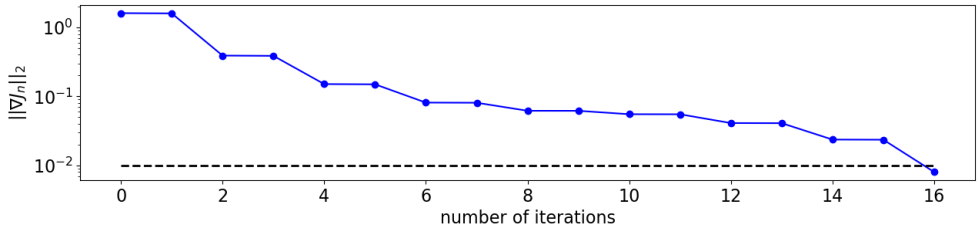
4.12. The estimated the values presented in table 4.5.



(a)



(b)



(c)

Figure 4.12: Plot of the errors $|a_n\text{-}a_{true}|$ of intensities (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)) on log scale at every iteration. Position was known. In the error plot, the green line corresponds to the lower left concentration, the blue to the lower right concentration and the red to the upper concentration. The dotted line in the gradient plot is the stopping tolerance. The initial guesses are 10, 0 and 5.

Table 4.5: True and estimated values of the three intensities, using DY. a_1 corresponds to the lower left concentration, a_2 corresponds to the lower right concentration and a_3 corresponds to the upper concentration. The initial guesses are 10, 0 and 5.

	a_1	a_2	a_3
True	3	5	6
DY	2.9018	5.0795	5.8927

Chapter 5

Time-dependent velocity field

In most cases, a velocity field will change with time. We thus study algorithm 2 with a time-dependent velocity field, obtained as follows. We start by solving the Poisson problem (4.1), to obtain the same velocity field as plotted in figure 4.3a. Then at each time step we rotate the velocity field, using the rotation matrix

$$W = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad \theta = \frac{2\pi t_i}{T}.$$

Using this velocity field, we simulate the concentration, plotted in figure 5.1. The white stars are the measurement locations. See Appendix A for the (x, y)-coordinates. $\Omega = [-1.5, 1.5] \times [-2, 2]$, discretized with triangles. The largest area of the triangles is 0.01 $[m^2]$. The concentration starts at the origin, with an intensity of $10/\mathrm{A}(\Delta)$. We simulate from $t = 0$ to t=3 using dt=0.15 as the time step and setting the diffusion coefficient to 10^{-9}. In this chapter, we look at the objective function (2.6) with weight w_m=$10(\mathrm{std}(u_m^*)+$ 0.1). The weights are plotted in figure 5.2. τ=0.0003 is the regularization parameter. The stopping criteria are $\|\nabla J + 2\tau \mathbf{x}_n\|_2 < 0.0065$, if we have not

converged after 20 iterations or if we are stuck. Both the restart criteria, and the gradient methods we test, are the same as in the previous chapter. For backtracking, the step size is initialized to 1, with $c_1 = 10^{-5}$. The scale factor is 0.015 if the next point is evaluated to be outside of the domain, and 0.8 otherwise.

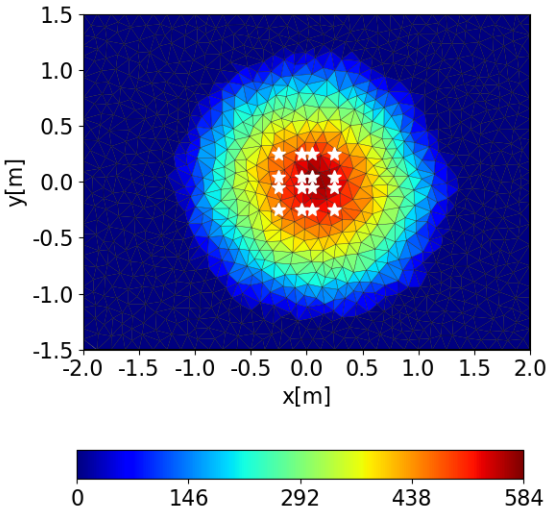


Figure 5.1: Snapshot of a concentration with time-dependent velocity field at $T = 3$. The white stars are the measurement locations.

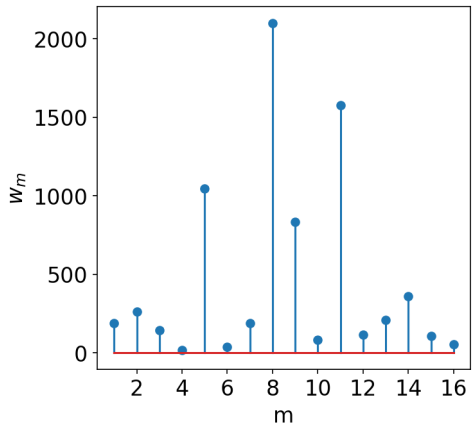


Figure 5.2: A plot of the weights used for the computations.

5.1 Fixed position

Just as in the previous chapter, we start by finding the flux, given that we know the correct position. Using 60 as the initial guess, we obtain the results presented in figure 5.3 and table 5.1 with DY.

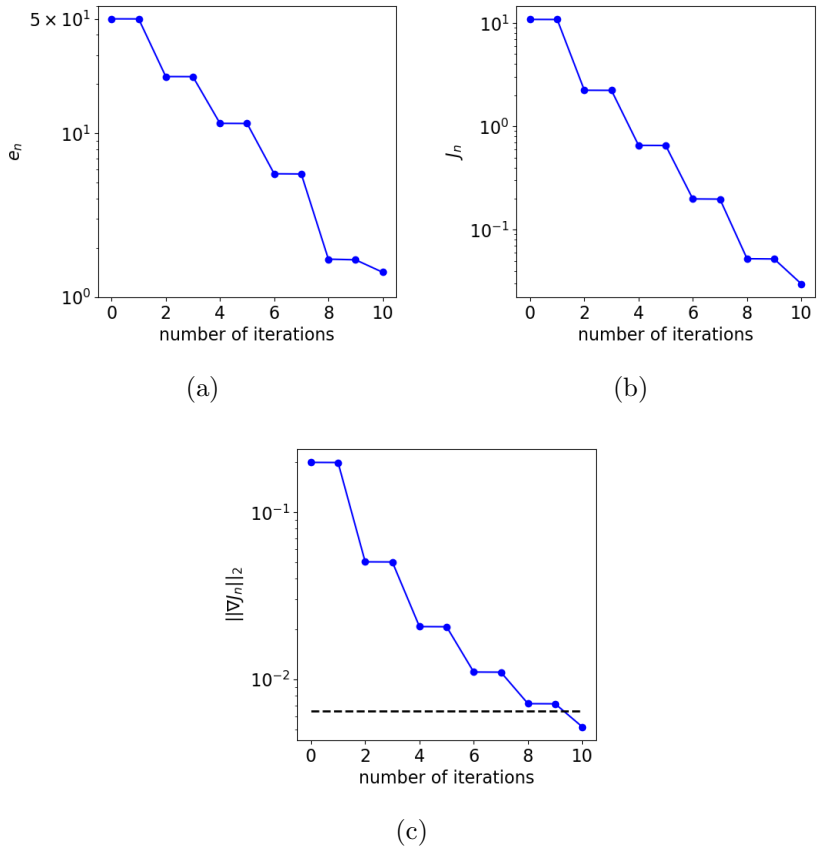


Figure 5.3: Plot of the error $|a_n\text{-}a_{true}|$ for intensity (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)) on log scale at every iteration. Position was known. The dotted line in the gradient plot is the stopping tolerance. The initial guess is 60.

Table 5.1: True value and estimated value of intensity, using DY. The initial guess is 60.

	True	DY
a	10	8.5817

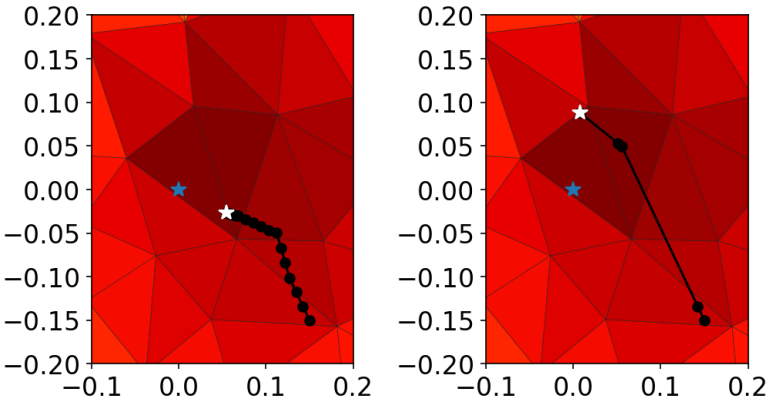
5.2 Fixed intensity

We now assume that the intensity is known and look to find the correct source position. Starting from $(x, y) = (0.15, -0.15)$, the estimated location found by GN and DY is shown in table 5.2. Figure 5.4 depicted the iterations in the domain. The blue star is the origin and the white star is where the method ends up. The values of the J_n and $\|\nabla J\|_2$ at each iteration are displayed in figure 5.5. We have not presented the outcome of SD, as it is not distinguishable from GN.

If the initial guess had been changed slightly to $(0.15, -0.2)$, then both SD and GN would get stuck at $(0.15, -0.19)$, while DY converges in three iterations.

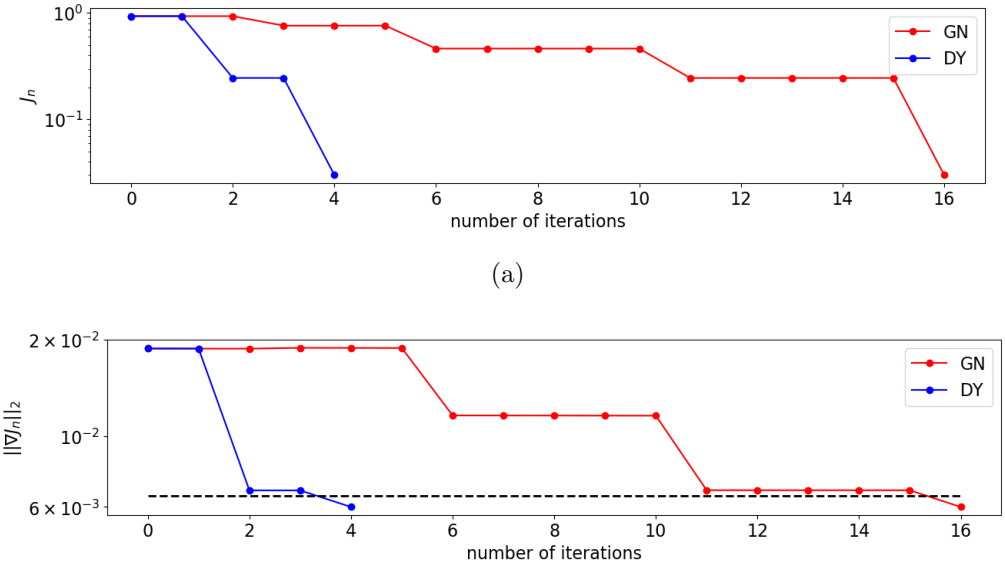
Table 5.2: True value and estimated values of source location, using GN and DY. The initial guess is $(-0.15, -0.15)$.

	True	GN	DY
(x, y)	(0,0)	(0.0542, -0.0269)	(0.0076, 0.0880)



(a) (b)

Figure 5.4: Iterations in the domain for GN (plot (a)) and DY (plot (b)). The white star is located where the particular method ends up, and the blue star is the true source position. The initial guess is $(0.15, -0.15)$.



(a)

(b)

Figure 5.5: Plot of the objective function (plot (a)) and of the 2-norm of the gradient (plot (b)) at every iteration on log scale. Intensity was known. The dotted line in the gradient plot is the tolerance. The initial guess was $(0.15, -0.15)$.

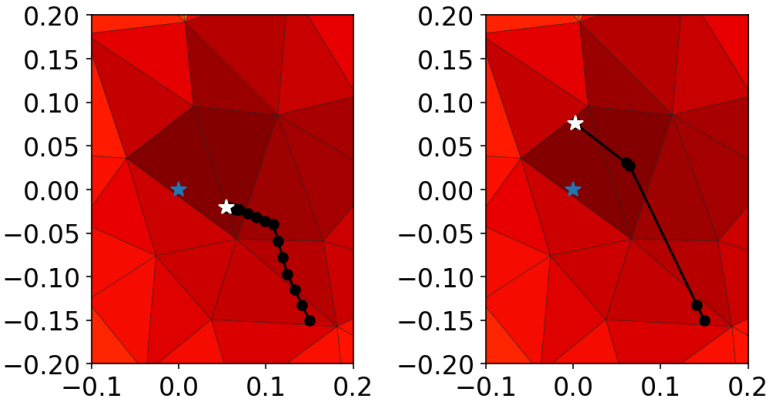
5.3 Position and intensity

We finally look to find both the intensity and position at the same time. Starting from the same point as in the previous section, figure 5.6 shows the iterations for both GN and DY in the domain. Compared to the last section, they are very similar. The estimated location is also very similar, see table 5.3.

The initial guess for intensity was 10.5, and figure 5.7a depicted the error for intensity at every step. Table 5.3 displayed the estimated intensity for both methods. For SD, both the estimated location and intensity are identical GN.

Values of J_n and $\|\nabla J\|_2$ are plotted in figure 5.7b and 5.7c, respectively. Just like position, they remained almost unchanged compared to the previous section. All values for SD,

Finally, just as in the preceding chapter, adding white Gaussian noise with standard deviation 2 to the measurements did not alter much of the outcome. The result only changed when we attempted to find the source location for DY, starting from $(0.15, -0.2)$. This time it got stuck.

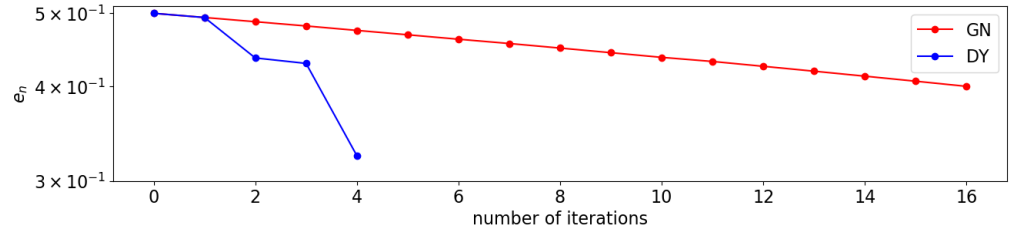


(a) GN (b) DY

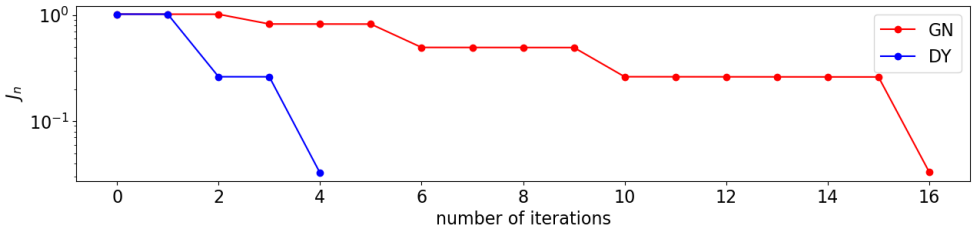
Figure 5.6: Iterations in the domain for GN (plot (a)) and DY (plot (b)). The white star is located where the particular method ends up, and the blue star is the true source position. The initial guess for position was $(0.15, -0.15)$, and 10.5 for intensity.

Table 5.3: True value and estimated values of intensity and source location, using GN and DY. The initial guess is 10.5 for intensity and $(-0.15, -0.15)$ for position.

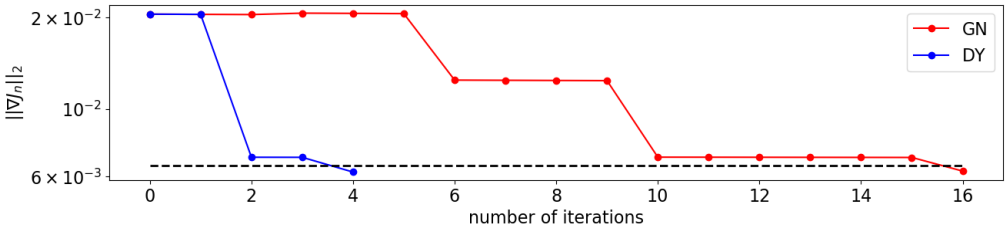
	True	GN	DY
a	10	10.4003	10.3204
$(x,\ y)$	$(0, 0)$	(0.0543, -0.0199)	(0.0022, 0.0756)



(a)



(b)



(c)

Figure 5.7: Plot of the error $|a_n - a_{true}|$ for intensity (plot (a)), of the objective function (plot (b)) and of the 2-norm of the gradient (plot (c)), at every step on log scale. Both position and intensity were unknown. The dotted line in the gradient plot is the stopping tolerance. The initial guess for position was $(0.15, -0.15)$, and 10.5 for intensity.

Chapter 6

Conclusion and further work

To summarize the work in this thesis, we have studied a source identification problem. By writing the problem as a PDE-constraint optimization problem, we derived the adjoint IBVP (3.25)-(3.27). This IBVP was used to compute the derivatives of the objective function (2.6) with respect to its parameters. These derivatives were then applied in a non-linear conjugate gradient method used to minimize (2.6) in order to estimate the parameters of a source function.

We used the steepest descent method, the Dai-Yuan method and the Gilbert-Nocedal method for the minimization on various test cases, using two different velocity fields. In the case of finding the intensity of the source, we saw in section 4.1 that the Dai-Yuan method outperformed the other methods. This was also the case in section 5.2, when we looked to find the source position. In section 4.2, on the other hand, all three methods used about the same number of iterations. Finally, we looked to find both intensity and location of the source in section 4.3 and section 5.3. Here, all three methods estimated the true location (i.e. triangle), but the intensity deviated from the true value. The only exception was the steepest descent

method, which in section 4.3 got pushed far away from the source location.

There are some problems that we have not adressed. One is how to pick a "good" initial guess for the gradient methods. A possible solution here, as suggested in [25] in a general context, is to use the solution obtained by a direct search method. Another important questions are how to determine where are the optimal measurement locations, and what are good choices of regularization and weight.

When using real data, a more suited PDE-solver has to be used. The pyGIMLI does not handle time-dependent source functions and time-dependent velocity fields. Moreover, one should scale the independent variables since gradient methods are poorly scaled ([22]). We used the simplest line search method. For real data this method could be used, but a more sophisticated line search method should be considered. Such a method could be found in chapter 3.5 in [22].

Theoretically, further studies would be to show existence and uniqueness of solution to the IBVP (2.1)-(2.3) with the source function (2.4) in the sense of source-type solutions.

Practically, it would be interesting to test some newer gradient methods; we refer to [14, 6] for an overview and analysis of new gradient methods. Moreover, comparing the gradient methods to quasi-Newton ([22, 25]) and Baysian methods ([2]) would be of interest.

Appendices

Appendix A

Coordinates for measurement locations

Table A.1: The coordinates of the measurement locations in figure 4.2a.

x	-1.2	-0.75	0.4	-2.15	0.4	0.9
y	0.2	0.75	-1.5	0.0	0.5	0.0
x	0.2	0.0	-1.75	-0.2	-2.2	0.0
y	-0.6	1.0	0.7	-1.1	-0.9	-0.1
x	-1.3	-0.6	-1.9	-0.5	0.6	1.0
y	1.1	-0.8	-1.7	1.5	-0.9	1.2
x	-1.3	0.2	-0.35	-0.35	-0.45	-1.3
y	-0.6	1.9	0.45	-0.9	-0.15	2.2
x	-2.2	0.0	-0.5	-1.0	-0.8	-0.7
y	1.6	-2.4	-1.3	-2.0	-0.5	-1.1

Table A.2: The coordinates of the measurement locations in figure 4.11

x	-1.0	1.1	0.05
y	-1.9	-1.3	0.65

Table A.3: The coordinates of the measurement locations in figure 5.1

x	-0.25	-0.05	0.05	0.25
y	-0.25	-0.25	-0.25	-0.25
x	-0.25	-0.05	0.05	0.25
y	-0.05	-0.05	-0.05	-0.05
x	-0.25	-0.05	0.05	0.25
y	0.05	0.05	0.05	0.05
x	-0.25	-0.05	0.05	0.25
y	0.25	0.25	0.25	0.25

Appendix B

Linear algebra and vector calculus

In this Appendix, we give a review of some facts and results from vector calculus that we have used in the thesis. The information is taken from [1], [25] and [28].

Let $\mathbf{x}, \mathbf{y}$ be two real vectors with entries x_i, y_i, $i = 1, 2, \ldots, n$, respectively. The inner product of $\mathbf{x}$ and $\mathbf{y}$ is

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y} = x_1 y_1 + \ldots x_n y_n,$$

where $\mathbf{x}^T$ is the transposed of $\mathbf{x}$. The Euclidean norm is the inner product

$$\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}.$$

For a function $f : \mathbb{R}^d \to \mathbb{R}$, its the gradient is the d-dimensional vector

$$\nabla f = \begin{bmatrix} f_{x_1} \\ f_{x_2} \\ \vdots \\ f_{x_d} \end{bmatrix},$$

where the subscripts on f in the bracket are the partial derivatives.

For a (unit) vector $\mathbf{u}$, and a point $\mathbf{p}$, *the directional derivative* of f in $\mathbf{u}$'s direction at the point $\mathbf{p}$, is given by

$$\lim_{t \to 0} \frac{f(\mathbf{p} + t\mathbf{u}) - f(\mathbf{p})}{t} = \mathbf{u} \cdot \nabla f(\mathbf{p}).$$

For any pair of functions g, f, *Green's first identity* states that

$$\iint_{\Omega} \nabla f \cdot \nabla g \, d\mathbf{x} + \iint_{\Omega} f \Delta g \, d\mathbf{x} = \int_{\partial\Omega} f \ (\mathbf{n} \cdot \nabla g) \, ds,$$

where $\partial\Omega$ is the boundary of the domain $\Omega \subset \mathbb{R}^2$ and $d\mathbf{x} = dxdy$. $\mathbf{n} \cdot \nabla g$ is the directional derivative in the outward normal direction on $\partial\Omega$, and ds is the element of arc length. $\Delta = \nabla \cdot \nabla$ is the Laplace operator. *Green's second identity* states that

$$\iint_{\Omega} g \, \Delta f - f \, \Delta g \, d\mathbf{x} = \int_{\partial\Omega} g \ (\mathbf{n} \cdot \nabla f) - f \ (\mathbf{n} \cdot \ \nabla g) \, ds,$$

holds for any pair of functions f, g.

Finally, let $\mathbf{f}$ denote a sufficiently smooth vector field. Then we have

$$\nabla \cdot (u\mathbf{f}) = \nabla u \cdot \mathbf{f} + u \ (\nabla \cdot \mathbf{f}), \tag{B.1}$$

for a smooth scalar field u. $\nabla \cdot f$ denotes the divergence of $\mathbf{f}$.

Appendix C

Function spaces

Definitions of the varoius function spaces in section 2.2. These definitions taken are from [8].

(i) $L^2(\Omega)$ consists of functions $f : \Omega \to \mathbb{R}$ such that

$$\|f\|_2 = \big(\int_\Omega |f|^2 dx \big)^{1/2} < \infty.$$

(ii) $H^{-1}(\Omega)$ is the dual space of the space $H_0^1(\Omega)$, the closure of $C_c^\infty(\Omega)$ in the Sobolev space $H^1(\Omega)$.

(iii) Let X be a real Banach space with norm $\|\cdot\|$. Then

- $L^2(0,T;X)$ consists of all strongly measurable functions $f : [0,T] \to X$, such that

$$\|f\|_{L^2(0,T;X)} = \left(\int_0^T \|f(t)\|^2 dt \right)^{1/2} < \infty$$

A function $f : [0,T] \to X$ is strongly measurable if there exist simple functions $s_k(t)$ which converges almost everywhere $0 \le t \le T$ to $f(t)$. And

– $C([0:T];\Omega)$ comprises all continuous function $f:[0,T]\to X$, with

$$\|f\|_{C(0,T;X)}=\max_{0\le t\le T}\|f(t)\|<\infty.$$

Bibliography

[1] R. Adams and C. Essex. *Calculus: A Comlpete Course*, chapter 9,12,13, 16. Pearson, 8 edition, 2013.

[2] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter estimation and inverse problems*. Elsevier, 3 edition, 2018.

[3] A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*, chapter 11. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2014.

[4] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*, chapter 1, pages 1–44. Springer New York, New York, NY, 2002.

[5] D. Coumou and S. Rahmstorf. A decade of weather extremes. *Nature climate change*, 2(7):491–496, 2012.

[6] Y.-H. Dai. Nonlinear conjugate gradient methods. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[7] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimization*, 10(1):177–182, 1999.

[8] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2 edition, 2010.

[9] H. Frøysa. Designs of a monitoring program in a varying marine environment. Master's thesis, University of Bergen, 2015.

[10] S. W. Funke and P. E. Farrell. A framework for automated pde-constrained optimisation, 2013.

[11] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992.

[12] A. Gjesteland. SBP-SAT schemes for hyperbolic problems. Master's thesis, University of Bergen, 2019.

[13] B. Gustafsson. *High Order Difference Methods for Time Dependent PDE*, chapter 2, pages 13–68. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[14] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[15] A. Hamdi. Inverse source problem in a 2d linear evolution transport equation: detection of pollution source. *Inverse Problems in Science and Engineering*, 20(3):401–421, 2012.

[16] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer Netherlands, Dordrecht, 2009.

[17] P. K. Kundu, I. M. Cohen, and D. R. Dowling. *Fluid Mechanics*, chapter 7, pages 296–299. Academic Press, 6 edition, 2016.

[18] X. Lai, Z. Ye, Z. Xu, M. H. Holmes], and W. H. Lambright]. Carbon capture and sequestration (ccs) technological innovation system in china: Structure, function evaluation and policy implication. *Energy Policy*, 50:635 – 646, 2012. Special Section: Past and Prospective Energy Transitions - Insights from History.

[19] L. Ling, Y. C. Hon, and M. Yamamoto. Inverse source identification for poisson equation. *Inverse Problems in Science and Engineering*, 13(4):433–447, 2005.

[20] G. Lu and H. Yin. Source-type solutions of heat equations with convection in several variables space. *Science China Mathematics*, 54(6):1145–1173, 2011.

[21] A. V. Mamonov and Y.-H. R. Tsai. Point source identification in nonlinear advection–diffusion–reaction systems. *Inverse Problems*, 29(3):035009, mar 2013.

[22] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, New York, NY, 2 edition, 2006.

[23] J. M. Pandolfi, S. R. Connolly, D. J. Marshall, and A. L. Cohen. Projecting coral reef futures under global warming and ocean acidification. *science*, 333(6041):418–422, 2011.

[24] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 11 2006.

[25] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*, chapter 7, pages 298–324. Springer, Berlin Heidelberg, 2 edition, 2007.

[26] C. Rücker, T. Günther, and F. M. Wagner. pyGIMLi: An open-source library for modelling and inversion in geophysics. *Computers and Geosciences*, 109:106–123, 2017.

[27] T. Sauer. *Numerical analysis*, chapter 4,5, pages 189–237, 257–259. Pearson Education Limited, 2 edition, 2014.

[28] W. A. Strauss. *Partial differential equations: An introduction*. John Wiley & Sons, 2 edition, 2008.

[29] K. Uchimoto, M. Nishimura, J. Kita, and Z. Xue. Detecting co2 leakage at offshore storage sites using the covariance between the partial pressure of co2 and the saturation of dissolved oxygen in seawater. *International Journal of Greenhouse Gas Control*, 72:130 – 137, 2018.