

# Sequential model correction for nonlinear inverse problems\*

Arttu Arjas<sup>†</sup>, Mikko J. Sillanpää<sup>†</sup>, and Andreas Hauptmann<sup>‡</sup>

**Abstract.** Inverse problems are in many cases solved with optimization techniques. When the underlying model is linear, first-order gradient methods are usually sufficient. With nonlinear models, due to nonconvexity, one must often resort to second-order methods that are computationally more expensive. In this work we aim to approximate a nonlinear model with a linear one and correct the resulting approximation error. We develop a sequential method that iteratively solves a linear inverse problem and updates the approximation error by evaluating it at the new solution. We separately consider cases where the approximation is fixed over iterations and where the approximation is adaptive. In the fixed case we show theoretically under what assumptions the sequence converges. In the adaptive case, particularly considering the special case of approximation by first-order Taylor expansion, we show that the sequence converges to a neighborhood of a local minimizer of the objective function with respect to the exact model. Furthermore, we show that with quadratic objective functions the sequence corresponds to the Gauss-Newton method. Finally, we showcase numerical results superior to the conventional model correction method. We also show, that a fixed approximation can provide competitive results with considerable computational speed-up.

**Key words.** Inverse problems, model approximation, optimization, nonlinear models

**AMS subject classifications.** 65K10, 65F20, 94A08, 47H10, 47A52

**1. Introduction.** Inverse problems appear in numerous places in mathematics, engineering and medicine [3, 23]. They deal with deducing cause from the observed effects (data). Mathematically, the problem is often written in the form

$$(1.1) \quad y = A(x),$$

where  $y \in Y$  is the measured data and  $x \in X$  the unknown cause. Here  $A : X \rightarrow Y$  is a (possibly) nonlinear model that describes how the measurements are linked to the cause. The inverse problem is then often solved by finding a cause that best matches the data when input into the model. Mathematically this is a variational optimization problem of the form

$$(1.2) \quad x^* = \arg \min_{u \in X} \{F(A(u), y) + \lambda R(u)\}$$

where the minimized objective function is a sum of functions  $F : Y \rightarrow \mathbb{R}_+$  and  $R : X \rightarrow \mathbb{R}_+$ . The first part measures the mismatch between the data and the model output (data fidelity), and the second part regularizes the solution. (Here,  $\lambda > 0$  is a regularization parameter to be adjusted for each data separately.) In this work we assume that both parts are given by convex functions, e.g. a norm. Given these assumptions, if the model is linear, the resulting objective

\*Submitted to the editors XYZ.

**Funding:** This work was funded by the Academy of Finland, under projects 338408, 346574, 326291 and the Centre of Excellence of Inverse Modelling and Imaging project 353093.

<sup>†</sup>Research Unit of Mathematical Sciences, University of Oulu, Finland. [arttu.arjas@oulu.fi](mailto:arttu.arjas@oulu.fi)

<sup>‡</sup>Research Unit of Mathematical Sciences, University of Oulu and Department of Computer Science, University College London, UK.

function is also convex. This means that it has a unique minimizer which can be efficiently found with first-order optimization techniques such as gradient descent or primal-dual methods which are usually computationally cheap to implement [4]. In this case the Jacobian (matrix consisting of the partial derivatives of the forward model) appearing in the gradient of the objective function is independent of the input of the model, and thus computationally cheap to evaluate repeatedly. However, if the model is nonlinear, the Jacobian depends on the input of the model and must be recomputed for every input. This becomes computationally inefficient particularly when the model is given in functional form and differentiation must be carried out by using numerical methods. Furthermore if the model is nonlinear, the objective function is generally nonconvex with multiple local minima. In this case first-order methods usually converge slowly since the gradient only carries information about the local steepness of the objective function.

The purpose of this work is to solve the nonlinear inverse problem with a linear approximation, unlocking the benefits of first-order convex optimization. As mentioned in the earlier paragraph, the benefits include faster convergence and speed-up in the computation. The approximation naturally creates an approximation error that we have to account for. Modeling errors can be efficiently corrected with the established approximation error method (AEM) [2, 14] and has been successfully applied in a wide area of inverse problems [6, 11, 24, 29, 35]. It is a linear correction that assumes that the approximation error has a Gaussian distribution. The mean vector and covariance matrix of the distribution must be estimated as a part of solving the inverse problem, for instance from training data. Naturally, AEM has trouble dealing with non-Gaussian approximation errors, which we also illustrate. To overcome this limitation, some recent works have proposed neural networks for model correction [12, 16, 20, 22, 33], in which case the correction is nonlinear. Neural networks succeed in the task but require training data that represent the solution space well enough. A related problem is model approximation using so-called surrogates [19, 39]. A surrogate model is an approximative model used often when the exact model is too slow to evaluate. The surrogate, modeled for example by neural networks or Gaussian processes, is trained with pairs of inputs and outputs of the true model and can greatly speed up for example Markov chain Monte Carlo (MCMC) computations requiring hundreds of thousands of model evaluations. However, surrogate-based models suffer from the curse of dimensionality as the number of training samples needed grows exponentially with the dimension of the parameter space.

We will particularly consider imaging applications in this work. A prototypical example of ill-posed inverse problem in imaging is deconvolution [23, 38]. The purpose of deconvolution is to restore an image degraded by some kernel. The kernel can be thought of as a filter that integrates local information of the image thereby reducing its quality. Convolution is also closely connected to diffusion processes, namely diffusion can be seen as convolution with a Gaussian kernel [1]. If the kernel does not depend on the image, diffusion is a linear operator. When the kernel is let depend on the image, the operator becomes nonlinear. This can be done for example with the Perona-Malik filter, where the diffusion strength depends on the magnitude of the image gradient [27]. Thus areas of the image with edges are diffused less than smoother areas.

To deal with the aforementioned problems, we propose a sequential model correction method. Starting from an initial point, we iteratively update the approximation error by

evaluating it at the current iterate and solve the inverse problem using the approximate model, eventually converging to a solution. The procedure locally linearizes and thus convexifies the variational problem. We investigate two different cases: one where the approximation is fixed over sequence iterations and one where the approximation adapts locally. The summary of the results of this paper is as follows:

- (i) For the fixed approximation, we derive the conditions needed for the convergence of the sequence.
- (ii) For the adaptive approximation, we show that under certain conditions on the approximation, taking small enough steps in the sequence always decreases the original objective function, i.e., the sequence converges to a neighborhood of a local minimizer.
- (iii) If the approximation is chosen as a first-order Taylor expansion of the accurate model and the data fidelity and regularization terms are both quadratic, we recover the Gauss-Newton method.

Finally we show with examples that the sequential model correction method outperforms AEM with various models, while being computationally efficient. While we have to be able to evaluate the accurate forward model, we need only few evaluations. Particularly, we show that a fixed approximation can yield similar quantitative results than the adaptive approximation, with a computational speed-up of up to factor 8.

This paper is organized as follows: in Section 2 we define notation, state the problem mathematically and introduce the approximation error method. In Section 3 we analyze the convergence of the method. The analysis is separated into fixed and adaptive cases. We also show the connection to the classic Gauss-Newton method. In Section 4 we introduce the models used in numerical experiments and discuss implementation details. The results are shown and discussed in Section 5. Finally, concluding remarks are given in Section 6.

**2. Problem statement.** Let us first establish notation and assumptions used throughout this paper unless otherwise stated.

Symbol	Meaning
$y$	Measured data
$x, u$	Unknown we want to reconstruct
$X, Y$	Hilbert spaces. Norm $\ \cdot\ _{X/Y}$ , inner product $\langle \cdot, \cdot \rangle_{X/Y}$
$A : X \rightarrow Y$	True nonlinear forward operator
$\tilde{A} : X \rightarrow Y$	Linear approximation of $A$
$F : Y \times Y \rightarrow \mathbb{R}_+$	Convex data fidelity functional
$R : X \rightarrow \mathbb{R}_+$	Convex regularization functional
$\lambda > 0$	Regularization parameter

We wish to solve the inverse problem

$$(2.1) \quad y = A(x) + e$$

for the unknown  $x$ , where  $e$  is noise. In the variational framework, solving the inverse problem amounts to solving the variational problem

$$(2.2) \quad x^* = \arg \min_{u \in X} \{F(A(u), y) + \lambda R(u)\}.$$

For solving the variational problem (2.2) one needs the Jacobian of  $A$ , as discussed earlier. Since  $A$  is nonlinear, the Jacobian depends on the input, and hence must be recomputed for each input which is time consuming particularly in iterative optimization algorithms. For this reason we wish to approximate  $A$  with a linear model  $\tilde{A}$ . The Jacobian of a linear model is independent of the input, making it computationally easier to handle. Writing (2.1) in terms of  $\tilde{A}$  yields

$$(2.3) \quad y = \tilde{A}x + A(x) - \tilde{A}x + e = \tilde{A}x + \varepsilon(x) + e.$$

We note that here the approximation by linear model creates an approximation error, denoted by  $\varepsilon(x)$ . Clearly this formulation of the model is still nonlinear, we have just moved the nonlinearity into  $\varepsilon(x)$ . Let us then assume that we have access to some initial reconstruction  $x_0 \in X$ . We can then write the model as

$$(2.4) \quad y \approx \tilde{A}x + \varepsilon(x_0) + e,$$

which is linear since  $x_0$  is known. This leads to the convex variational problem

$$(2.5) \quad x^* = \arg \min_{u \in X} \left\{ F(\tilde{A}u, y - \varepsilon(x_0)) + \lambda R(u) \right\},$$

which gives us a local reconstruction depending on  $x_0$ . From here it is natural to expand this construction into a sequence

$$(2.6) \quad x_{k+1} = S(x_k) = \arg \min_{u \in X} \left\{ F(\tilde{A}u, y - \varepsilon(x_k)) + \lambda R(u) \right\}.$$

We emphasize that updating the sequence, i.e., solving a linearized and thus convex optimization problem can be done efficiently with first-order optimization methods.

In the following we aim to:

- (i) Analyze the sequence (2.6) theoretically from a model correction perspective. We derive conditions needed for the convergence of the sequence. We also draw connections to existing methods for nonconvex optimization, i.e., with certain choices for the approximate model and the data fidelity and regularization functionals, we obtain the Gauss-Newton algorithm.
- (ii) Show that the sequential model correction method delivers superior results compared to the conventional method.

### 2.1. Approximation error method and non-Gaussianity of the approximation error.

AEM has traditionally been used for model correction in inverse problems. It exploits the normal distribution to integrate the approximation error out of the model. It assumes that  $\varepsilon(x) \sim \mathcal{N}(\mu_\varepsilon, \Sigma_\varepsilon)$ , where  $\mu_\varepsilon$  and  $\Sigma_\varepsilon$  are estimated for instance from a training data set

$\{x_i\}_{i=1}^N$  [14, 20]. The accurate and approximate model are applied to each data point such that  $\varepsilon_i = A(x_i) - \tilde{A}x_i$ . Then the estimates for the mean vector and covariance matrix are

$$(2.7) \quad \hat{\mu}_\varepsilon = \frac{1}{N} \sum_{i=1}^N \varepsilon_i, \quad \hat{\Sigma}_\varepsilon = \frac{1}{N-1} \sum_{i=1}^N (\varepsilon_i - \hat{\mu}_\varepsilon)(\varepsilon_i - \hat{\mu}_\varepsilon)^T.$$

Furthermore, the noise  $e$  is assumed independently Gaussian with zero-mean and variance  $\sigma^2$ . Thus we have  $\text{cov}(\varepsilon + e) = \Sigma_\varepsilon + \sigma^2 I$ , assuming mutual independence of the terms. This allows us to write the variational problem as

$$(2.8) \quad x^* = \arg \min_{u \in \mathbb{R}^n} \left\{ \frac{1}{2} \|L^{-1}(\tilde{A}x - y + \mu_\varepsilon)\|_Y^2 + \lambda R(u) \right\},$$

where  $L$  is the Cholesky factor of  $\Sigma_\varepsilon + \sigma^2 I$  such that  $\Sigma_\varepsilon + \sigma^2 I = L^T L$ . We note that here we need to assume that the data fidelity is given as a squared  $L^2$ -norm.

AEM specifically assumes that the approximation error has a Gaussian distribution. In this case multiplying the data with the inverse of the Cholesky factor whitens the noise, making it identically Gaussian, which justifies the use of squared  $L^2$ -data fidelity. However, non-Gaussian errors arise especially when trying to correct nonlinear models. This can be seen by assuming a Gaussian distribution for the unknown, i.e.,  $x \sim \mathcal{N}(m, C)$ , and then looking at the distribution of the approximation error. When approximating a linear model with a linear model, the distribution of the approximation error  $(A - \tilde{A})x$  is still Gaussian with mean  $(A - \tilde{A})m$  and covariance  $(A - \tilde{A})C(A - \tilde{A})^T$ . With nonlinear  $A$  this is no longer the case since the mean is not a linear function of  $m$ , meaning it would be different for all  $x \in X$ . Similarly the covariance would involve the Jacobian of  $A$ , which is different depending on the point in  $X$  we are at.

**3. Convergence of the sequence.** We split the theoretical analysis into two parts. In the first part we assume that the approximation is fixed, i.e. it does not depend on the sequence iteration index  $k$ . In the second part we let the approximation depend on  $k$ . In the fixed case, we state the conditions needed for the convergence of the sequence. For the adaptive case, we consider the special case of local linear approximation at  $x_k$  given by the first-order Taylor expansion. In that case we can show that we are minimizing the original objective function given by the nonlinear model, and hence convergence to a neighborhood of local minimizer of it. The obtained result readily extends to other approximations that provide a descent direction.

**3.1. Fixed approximation.** In the case of fixed approximation we investigate the convergence conditions of the sequence based on fixed point iterations. We recall that an iterated function is defined by a composition  $S \circ \dots \circ S$  for  $S : X \rightarrow X$ . Then, the Banach fixed-point theorem states that if  $S$  is contractive, i.e.,  $K$ -Lipschitz with  $K < 1$ , the iterations will converge to a unique fixed point. We will use this theorem to state general conditions under which the sequence (2.6) converges. We will start with simple cases to gain intuition into the subject and work our way to more general cases of the functional  $F(A(u), y) + \lambda R(u)$ .

We start by looking at a simple invertible linear system and the behavior when we approximate the exact operator with another invertible linear operator.

**Theorem 3.1 (Linear and invertible operators).** *Let both  $A : X \rightarrow Y$  and  $\tilde{A} : X \rightarrow Y$  be linear and invertible and  $R = 0$ . Furthermore, let the operator norm of  $I - \tilde{A}^{-1}A$  be smaller than one. Then sequence (2.6) converges to  $x^* = A^{-1}y$ .*

*Proof.* We have  $x_{k+1} = S(x_k) = \tilde{A}^{-1}(y - \varepsilon(x_k))$ . Hence for  $x_1, x_2 \in X$ ,

$$\begin{aligned} \|S(x_1) - S(x_2)\|_X &= \|x_1 - \tilde{A}^{-1}Ax_1 - x_2 + \tilde{A}^{-1}Ax_2\|_X \\ &= \|(I - \tilde{A}^{-1}A)(x_1 - x_2)\|_X \\ &\leq K\|x_1 - x_2\|_X, \end{aligned}$$

where the Lipschitz constant  $K$  is the operator norm of  $I - \tilde{A}^{-1}A$ . Thus by Banach fixed-point theorem,  $S$  is a contraction and admits a unique fixed point. Let  $x^*$  be a fixed point of  $S$ . Then

$$\begin{aligned} (3.1) \quad x^* &= \tilde{A}^{-1}(y - Ax^* + \tilde{A}x^*) \\ &\iff x^* = A^{-1}y \end{aligned} \quad \blacksquare$$

The result essentially means that if  $\tilde{A}^{-1}A$  is close enough to the identity, that is,  $A$  is close to  $\tilde{A}$ , the sequence converges to the exact solution of the original problem.

We now examine a case where we need regularization to make the inverse problem uniquely solvable. In particular, we still assume that both  $A$  and  $\tilde{A}$  are linear and both the data fidelity and regularizer are given as quadratic functionals. This yields the sequence

$$(3.2) \quad x_{k+1} = \arg \min_{u \in X} \left\{ \frac{1}{2} \|\tilde{A}u - y + \varepsilon(x_k)\|_Y^2 + \frac{\lambda}{2} \|u\|_X^2 \right\}.$$

**Theorem 3.2 (Linear operators with Tikhonov regularization).** *Let both  $A : X \rightarrow Y$  and  $\tilde{A} : X \rightarrow Y$  be linear,  $F(\tilde{A}u, y - \varepsilon(x_k)) = \frac{1}{2} \|\tilde{A}u - y + \varepsilon(x_k)\|_Y^2$ ,  $R(u) = \frac{1}{2} \|u\|_X^2$  and  $T = (\tilde{A}^* \tilde{A} + \lambda I)^{-1} \tilde{A}^*$ . Moreover, let the operator norm of  $T(A - \tilde{A})$  be less than one. Then sequence (2.6) converges to*

$$x^* = \left( I + T(A - \tilde{A}) \right)^{-1} Ty.$$

*Proof.* In this case we have  $x_{k+1} = S(x_k) = Ty - T\varepsilon(x_k)$ . Hence for  $x_1, x_2 \in X$ ,

$$\begin{aligned} (3.3) \quad \|S(x_1) - S(x_2)\|_X &= \|T\varepsilon(x_1) - T\varepsilon(x_2)\|_X \\ &= \|TAx_1 - T\tilde{A}x_1 - TAx_2 + T\tilde{A}x_2\|_X \\ &= \|T(A - \tilde{A})(x_1 - x_2)\|_X \\ &\leq K\|x_1 - x_2\|_X, \end{aligned}$$

where the Lipschitz constant  $K$  is the operator norm of  $T(A - \tilde{A})$ . Thus, by Banach fixed-point theorem,  $S$  is a contraction and admits a unique fixed point. Let  $x^*$  be a fixed point of  $S$ . Then,

$$\begin{aligned} (3.4) \quad x^* &= Ty - T(A - \tilde{A})x^* \\ &\iff x^* = \left( I + T(A - \tilde{A}) \right)^{-1} Ty \end{aligned} \quad \blacksquare$$

Now the solution is the minimizer of the Tikhonov functional with the approximate operator multiplied by a correction term  $(I + T(A - \tilde{A}))^{-1}$ . Clearly the correction term is just identity when  $A = \tilde{A}$ .

Let us make a few remarks. In the previous cases we required a linear operator to be contractive. This can be verified for example through power method [8]. For nonlinear operators it is harder to verify. However, there are three heuristics we can use to enforce contractivity in that case as well.

- (i) We note that in the linear cases the Lipschitz constant (largest singular value) depends on  $A - \tilde{A}$ . The better the approximation is, the more likely the sequence converges.
- (ii) In the Tikhonov regularized case, the Lipschitz constant depends on  $T$ , which in turn depends on the regularization parameter  $\lambda$ . In particular, increasing  $\lambda$  decreases the Lipschitz constant. This is because the singular values of  $\tilde{A}^* \tilde{A} + \lambda I$  increase with  $\lambda$ , while the singular values of  $(\tilde{A}^* \tilde{A} + \lambda I)^{-1}$  decrease with  $\lambda$ .
- (iii) We can define a damped sequence as

$$(3.5) \quad x_{k+1} = S^\delta(x_k) = \delta_k S(x_k) + (1 - \delta_k)x_k,$$

where  $\delta \in \mathbb{R}$ . The damped sequence has the same fixed point as the original sequence, if it exists. However, now the Lipschitz constant of the update rule depends on  $\delta$ . In theory, it is possible to find a value of  $\delta$  that minimizes the Lipschitz constant at the fixed point. This trick is used for example in the Babylonian method for computing square roots [17]. Using this method requires knowledge of the solution which is not available in practice. It is still possible to experiment with different values and examine the sequence's behaviour.

In the nonlinear case with a general regularizer it is harder to make a general statement. We can still investigate the Lipschitz properties of the map  $S$  and recall that convergence occurs when the Lipschitz constant of  $S$  is smaller than one. To do that, we need some more assumptions. In particular, we need  $S$  to map between bounded sets. This is an appropriate assumption in imaging applications where we can assume that the pixels of the image take values between zero and one thus making the image space bounded. The boundedness can be enforced for example by projected gradient descent or primal-dual methods where the image is simply projected to a convex and bounded set at every iteration.

**Theorem 3.3.** *Let  $A : X \rightarrow Y$  be a nonlinear and  $\tilde{A} : X \rightarrow Y$  a linear operator. Also, let  $S : B \rightarrow B$ , where  $B$  is a bounded subspace of  $X$ . Then the map  $S$  is Lipschitz.*

*Proof.* Since  $B$  is bounded, we have  $d(x_1, x_2) < r$  for all  $x_1, x_2 \in B$  and for some  $r > 0$ . Also  $d(S(x_1), S(x_2)) < r$ , since  $S(x) \in B$  for all  $x \in B$ . Therefore there exists a constant  $K > 0$  such that  $Kd(x_1, x_2) \geq r$ . This in turn implies that  $d(S(x_1), S(x_2)) \leq Kd(x_1, x_2)$ , which means that  $S$  is  $K$ -Lipschitz. ■

We emphasize that it is difficult to tell whether some approximation  $\tilde{A}$  is good enough with respect to the exact model  $A$  to yield a convergent sequence. To ensure convergence in practice however, one could evaluate the objective function  $F(A(u), y) + \lambda R(u)$  at each iterate  $x_k$  and terminate the sequence when the objective function can no longer be decreased.



**3.2. Adaptive approximation.** We may also let the approximation  $\tilde{A}$  depend on  $k$ , i.e. we let the approximation  $\tilde{A}_k$  change at every step of the sequence. Then the sequence is given in general form as

$$(3.6) \quad x_{k+1} = \delta_k \arg \min_{u \in X} \left\{ F(\tilde{A}_k u, y - \varepsilon(x_k)) + \lambda R(u) \right\} + (1 - \delta_k)x_k,$$

with step size  $\delta_k$ . This case can not be analyzed with the fixed-point theory since the mapping changes at each iteration. Here, one could consider a variety of approximations, but we will specifically focus on the case where the approximation is given as a first-order Taylor expansion centered at  $x_k$ . In that case,  $\tilde{A}_k$  is the Jacobian of  $A$  evaluated at  $x_k$ .

We note, that this choice for the adaptive approximation yields a sequence similar to the successive linearized and regularized Gauss-Newton method, see [21,36] for recent applications. Nevertheless, we emphasize that this is a special case of the sequence formulation in (3.6) and as such naturally serves as a reference method for the fixed approximation.

**3.2.1. Approximation by Taylor expansion.** Let us now consider local linear approximations by Taylor expansion. That is, the approximation is of the form  $\tilde{A}x = A(x') + J(x')(x - x')$ , where  $x' \in X$  is given and  $J(x)$  is the Jacobian of  $A$  evaluated at  $x$ . A convenient choice is to choose  $x'$  as the current element of the sequence, yielding

$$(3.7) \quad x_{k+1} = \delta_k \arg \min_{u \in X} \{ F(J_k u, y - A(x_k) + J_k x_k) + \lambda R(u) \} + (1 - \delta_k)x_k,$$

where we denote  $J(x_k) = J_k$  for clarity. We note, that this scheme is also related to surrogate approaches for nonconvex optimization, for instance the majorization-minimization algorithm [18], which sequentially constructs surrogates that majorize the nonconvex function and minimizes the surrogates. Likewise, the adaptive sequential method constructs a convex surrogate of the nonconvex function at  $x_k$ . The difference is that the surrogate does not necessarily majorize the non-convex function. These surrogates are sequentially minimized, eventually reaching a neighborhood of local minimizer of the nonconvex function. We state this result in the following theorem.

**Theorem 3.4 (Descend property of the adaptive sequence).** *Let  $A : X \rightarrow Y$  be a nonlinear operator,  $F : Y \times Y \rightarrow \mathbb{R}_+$  a convex data fidelity term and  $R : X \rightarrow \mathbb{R}_+$  a convex regularization functional. Moreover, let  $A$ ,  $F$  and  $R$  also be continuously differentiable. Define  $L(u) = F(A(u), y) + \lambda R(u)$ ,  $L_k^s(u) = F(J_k u, y - A(x_k) + J_k x_k) + \lambda R(u)$  and let  $\nabla L(x_k) \neq 0$ , i.e.,  $x_k$  is not a stationary point of  $L$ . Then, for small enough  $\delta_k \in [0, 1]$ , the sequence (3.7) converges to a neighborhood of a local minimizer of  $L$ .*

*Proof.* Since  $\tilde{A}_k = J_k$  is linear and  $F$  and  $R$  are convex,  $L_k^s$  is a convex surrogate of  $L$  centered at  $x_k$ . We need to show that  $L(x_{k+1}) < L(x_k)$ . It suffices to show  $\langle \nabla L(x_k), x_{k+1} - x_k \rangle_X$  is negative which means that the vector pointing towards  $x_{k+1}$  from  $x_k$  is a descent direction of  $L$  at  $x_k$ . By the convexity of  $L_k^s$ , we have

$$L_k^s(x_k) + \langle \nabla L_k^s(x_k), u - x_k \rangle_X = L_k^s(x_k) + \langle \nabla L(x_k), u - x_k \rangle_X \leq L_k^s(u)$$

for all  $u \in X$ , since  $\nabla L(x_k) = \nabla L_k^s(x_k) = J_k^* \nabla F(A(x_k), y) + \lambda \nabla R(x_k)$ . It directly follows that  $\langle \nabla L(x_k), x_{k+1} - x_k \rangle_X$  is negative since  $L_k^s(x_{k+1}) < L_k^s(x_k)$  ( $x_{k+1} - x_k$  is also a descent direction for  $L_k^s$  at  $x_k$ ). ■



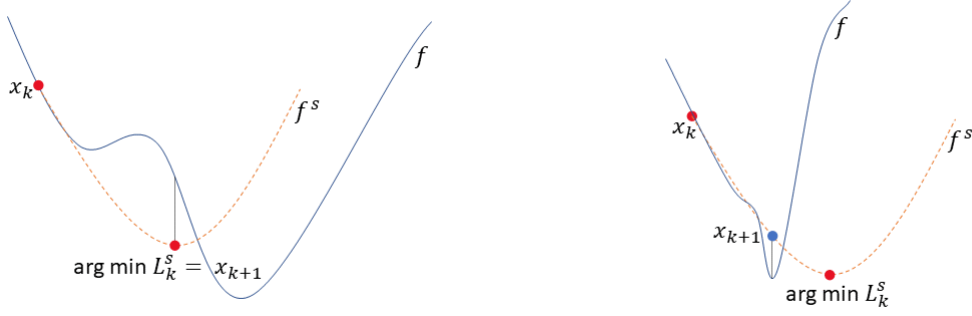


Figure 3.1: Illustration of the surrogate function that results from the local linear approximation. We may be able to get past some local minima, as shown in the picture on the left. The picture on the right shows that not controlling the step size might actually result in increasing the function value.

We emphasize, that the adaptive approximation for the sequence (3.6), does not necessarily need to be the Jacobian of the exact model. In this case, we can formulate a corollary with a general condition to obtain similarly local convergence.

**Corollary 3.5.** *Let  $\tilde{A}_k : X \rightarrow Y$  be a linear operator such that the iterates of the sequence (3.6) satisfy  $\langle \nabla L(x_k), x_{k+1} - x_k \rangle_X < 0$ , that is,  $x_{k+1} - x_k$  is a descent direction for the functional  $L$  at  $x_k$ . Then the sequence (3.6) converges to a neighborhood of a local minimizer of  $L$ .*

Figure 3.1 illustrates the convex surrogate obtained with the local linear approximation. It might let us escape some local minima. However, taking too long steps may end up increasing the value of the function we try to minimize. The optimal step size can be chosen for example by line search, that is, we choose such  $\delta$  that minimizes the objective function:

$$(3.8) \quad \delta_k = \arg \min_{\delta^* \in [0,1]} L(\delta^* \arg \min_{u \in X} \{L_k^s(u)\} + (1 - \delta^*)x_k)$$

**3.2.2. Connection to the Gauss-Newton method.** The Gauss-Newton method was originally designed for solving nonlinear least-squares problems [25]. It is an iterative optimization method involving linearization of the nonlinear model at every iterate. In inverse problems, it is most often used to solve variational problems of the form

$$(3.9) \quad x^* = \arg \min_{u \in X} \left\{ \frac{1}{2} \|A(u) - y\|_Y^2 + \lambda R(u) \right\},$$

where  $R$  must be twice continuously differentiable. The updating step is given as [25, 30]

$$(3.10) \quad x_{k+1} = x_k + \delta_k (J_k^* J_k + \lambda \nabla^2 R_k)^{-1} [J_k^* (y - A(x_k)) - \lambda \nabla R_k],$$

where  $J_k$  is the Jacobian of  $A$  evaluated at  $x_k$ ,  $\nabla R_k$  is the gradient and  $\nabla^2 R_k$  is the Hessian of  $R$  evaluated at  $x_k$  and  $\delta_k$  is a step size chosen with line search. It turns out that with certain choices the adaptive sequential method is equivalent to the Gauss-Newton algorithm. In particular, we have to always choose  $F$  as the  $L^2$  norm. Then, if we choose  $R = 0$ , requiring that  $J_k^* J_k$  is invertible, the adaptive sequential update becomes

$$(3.11) \quad \begin{aligned} x_{k+1} &= \delta_k (J_k^* J_k)^{-1} J_k^* (y - A(x_k) + J_k x_k) + (1 - \delta_k) x_k \\ &= x_k + \delta_k (J_k^* J_k)^{-1} J_k^* (y - A(x_k)), \end{aligned}$$

which is exactly the same as Gauss-Newton update. Further, if we choose  $R(u) = \frac{1}{2} \|u\|_X^2$ , the adaptive sequential update is

$$(3.12) \quad x_{k+1} = \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* (y - A(x_k) + J_k x_k) + (1 - \delta_k) x_k.$$

The connection to Gauss-Newton is a bit more tricky to see here. The Gauss-Newton update can be written as

$$(3.13) \quad \begin{aligned} x_{k+1} &= x_k + \delta_k (J_k^* J_k + \lambda I)^{-1} [J_k^* (y - A(x_k)) - \lambda x_k] \\ &= (I - \lambda \delta_k (J_k^* J_k + \lambda I)^{-1}) x_k + \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* (y - A(x_k)) \\ &= (1 - \delta_k) x_k + \delta_k J_k^* (J_k J_k^* + \lambda I)^{-1} J_k x_k + \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* (y - A(x_k)) \\ &= (1 - \delta_k) x_k + \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* J_k x_k + \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* (y - A(x_k)) \\ &= \delta_k (J_k^* J_k + \lambda I)^{-1} J_k^* (y - A(x_k) + J_k x_k) + (1 - \delta_k) x_k, \end{aligned}$$

which is the same as the adaptive sequential update in (3.12). The third row in the above equation follows by Woodbury formula [10]

$$(3.14) \quad (J_k^* J_k + \lambda I)^{-1} = \lambda^{-1} I - \lambda^{-2} J_k^* (I + \lambda^{-1} J_k J_k^*)^{-1} J_k$$

and the fourth row by push-through identity [13]

$$(3.15) \quad J_k^* (J_k J_k^* + \lambda I)^{-1} = (J_k^* J_k + \lambda I)^{-1} J_k^*.$$

In general, the adaptive sequential method and Gauss-Newton algorithm are the same for quadratic objective functions. The reason for this is that one step of Gauss-Newton is enough to minimize a quadratic function.

**3.3. Non-differentiable case.** In the previous section we required the data fidelity  $F$  and regularization term  $R$  to be continuously differentiable functions. However, in principle, the sequential formulation allows the use of any data fidelity and regularization term. In practice, and as we will show in the experiments, we observe good performance for non-differentiable  $L^1$  data fidelity and total variation (TV) regularization. This implies that the theory could be modified to allow for non-differentiable objective functions and possibly non-convex data fidelity or regularizer. As these are not the focus of this study, we leave the analysis of other cases for future studies.

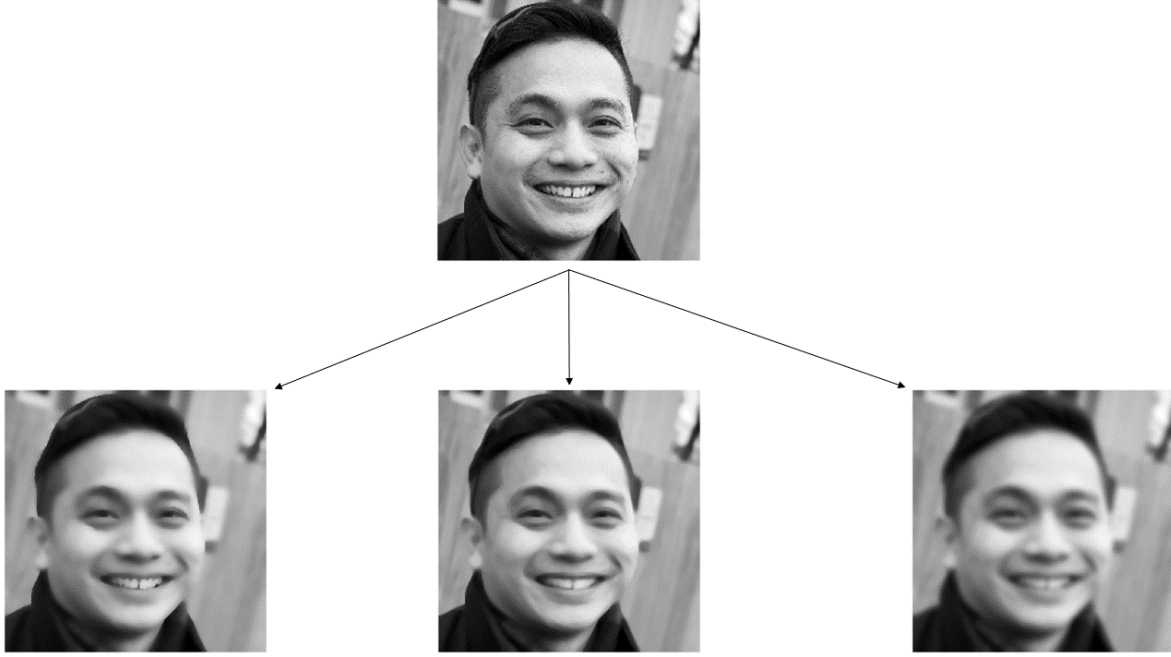


Figure 3.2: Illustration of how the operators considered in this work distort the ground truth image (top). Nonlinear diffusion (NLD) (bottom left), curvature flow (CF) (bottom middle) and linear diffusion (LD) (bottom right).

**4. Models and implementation.** We test the sequential correction method with diffusion-type operators that have been extensively used in imaging. A general diffusion operator  $A : x_0 \mapsto x_T$  on  $\mathbb{R}^d$  for a fixed time interval  $(0, T]$  is defined by the partial differential equation

$$(4.1) \quad \begin{cases} \partial_t x &= \nabla \cdot (\gamma(x) \nabla x) \\ x &= x_0 \text{ when } t = 0, \end{cases}$$

for diffusivity  $\gamma$ . If we let  $\gamma = 1$ , the operator becomes linear and the model corresponds to convolution with a Gaussian kernel. We can also consider a model with varying diffusivity. One possibility is to use the Perona-Malik filter such that  $\gamma(x) = (1 + |\nabla x|^2 / \kappa^2)^{-1}$ , where  $\kappa > 0$  is a contrast parameter [27]. Numerically the operator can be implemented by explicit iterative time-stepping algorithm. Starting from initial value  $x_0$ , the  $k$ -th iteration is defined as

$$(4.2) \quad x_{k+1} = x_k + \delta t \nabla \cdot \gamma(x_k) \nabla x_k,$$

with step size  $\delta t$ . We also consider a related curvature flow operator. On  $\mathbb{R}^d$  and fixed time interval  $(0, T]$  the operator is defined as [38]

$$(4.3) \quad \begin{cases} \partial_t x &= |\nabla x| \nabla \cdot \left( \frac{\nabla x}{|\nabla x|} \right) \\ x &= x_0 \text{ when } t = 0. \end{cases}$$

Similarly as with the diffusion operator, the curvature flow operator can be numerically implemented with iterative time-stepping algorithm

$$(4.4) \quad x_{k+1} = x_k + \delta t (|\nabla x| + \psi) \nabla \cdot \left( \frac{\nabla x_k}{|\nabla x_k| + \psi} \right)$$

with step size  $\delta t$  and threshold  $\psi$  for differentiability and stability by avoiding division with a number too close to zero. The models are referred to as NLD, CF and LD, for nonlinear diffusion, curvature flow and linear diffusion, respectively. Figure 3.2 illustrates the effect of the three models when applied on an image.

**4.1. Numerical experiments.** We used the developed sequential model correction method to restore images distorted by the nonlinear diffusion and curvature flow operators. For the fixed method we used a linear diffusion operator (constant diffusivity) to approximate the nonlinear operators in both cases. For the adaptive method we used the Jacobian of the nonlinear operator evaluated at each iterate as an approximation. We set the contrast parameter in Perona-Malik filter to  $\kappa = 0.1$  and the threshold parameter in the curvature flow model to  $\psi = 0.001$ . The step size  $\delta t$  was set to 0.1 and number of steps to 15 in all time-stepping algorithms. Either 3 % of Gaussian noise or 4 % of impulse noise was added to the distorted image. We numerically evaluated the reconstruction quality over a batch of 32 images, measuring the correspondence to ground truth images using peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [37]. We also investigated empirically the convergence of the methods by computing the value of the data fidelity at each iteration of the sequence.

We compared our methods to the case where the approximation error was not dealt with, i.e., the reconstruction was computed using only the approximate model. We also compared our methods to AEM, where the approximation error is assumed to have a Gaussian distribution. The evaluation was done separately for the nonlinear diffusion and curvature flow models with Gaussian and impulse (salt and pepper) noise which assigns random pixels a value zero or one [9]. For the Gaussian noise case we assumed squared  $L^2$  data fidelity and for the impulse noise case we assumed  $L^1$  data fidelity. In all experiments we used TV regularization which penalizes the absolute value of the image gradient, favoring piecewise constant reconstructions [28]. The regularization parameter was chosen separately for different methods by computing reconstructions for a test image with different values of  $\lambda$  and selecting the value that yielded the highest SSIM.

**4.2. Implementation.** The algorithms for sequential model correction and AEM were implemented with Python. The algorithm for sequential model correction with squared  $L^2$ - or  $L^1$ -data fidelity and TV regularization is presented in Algorithm 4.1. The subproblems requiring convex optimization were solved with primal-dual methods [7, 32]. The Jacobians for local linear approximations were computed via Jacobian-vector and vector-Jacobian products using the autograd library in Pytorch [26]. The computations were performed with a workstation with two 2.20 GHz processors and Nvidia Quadro P4000 GPU.

**4.3. Data.** For testing our methods we used the FFHQ dataset [15]. The dataset consists of 70000 color images of aligned and cropped faces of size  $1024^2$ . The data was preprocessed by

---

**Algorithm 4.1** Sequential model correction with primal-dual method for  $L^2/L^1$  data fidelity and TV regularization

---

```

1: Require  $x_0, \lambda$ 
2:  $u_0 \leftarrow x_0, \bar{u}_0 \leftarrow x_0, p_0 \leftarrow 0, q_0 \leftarrow 0, k \leftarrow 0, \theta \leftarrow 1$ 
3: while sequence not converged do
4:    $K_k \leftarrow \|(\tilde{A}_k, \nabla)\|_{\text{op}}$ 
5:    $\tau_k \leftarrow 1/K_k$ 
6:    $\sigma_k \leftarrow 1/K_k$ 
7:    $\varepsilon(x_k) = A(x_k) - \tilde{A}x_k$ 
8:    $t \leftarrow 0$ 
9:   while primal-dual not converged do
10:     $r_{t+1} = \tilde{A}_k \bar{u}_t - y + \varepsilon(x_k)$ 
11:     $p_{t+1} \leftarrow (p_t + \sigma_k r_{t+1}) / (1 + \sigma_k)$   $\triangleright L^2$ 
12:     $p_{t+1} \leftarrow (p_t + \sigma_k r_{t+1}) / \max(1_Y, |p_t + \sigma_k r_{t+1}|)$   $\triangleright L^1$ 
13:     $q_{t+1} \leftarrow \lambda(q_t + \sigma_k \nabla \bar{u}_t) / \max(\lambda 1_X, |q_t + \sigma_k \nabla \bar{u}_t|)$ 
14:     $u_{t+1} \leftarrow u_t - \tau_k \tilde{A}_k^* p_{t+1} + \tau_k \text{div}(q_{t+1})$ 
15:     $\bar{u}_{t+1} \leftarrow u_{t+1} + \theta(u_{t+1} - u_t)$ 
16:     $t \leftarrow t + 1$ 
17:   end while
18:   Perform line search to find  $\delta_k$ 
19:    $x_{k+1} \leftarrow (1 - \delta_k)\bar{u}_t + \delta_k x_k$ 
20:    $k \leftarrow k + 1$ 
21:    $\bar{u}_0 \leftarrow \bar{u}_t, u_0 \leftarrow u_t, p_0 \leftarrow p_t, q_0 \leftarrow q_t$ 
22: end while
23: return  $x_k$ 

```

---

	NLD (Gaussian)		NLD (Impulse)	
	PSNR	SSIM	PSNR	SSIM
No correction	26.96 (2.58)	0.87 (0.039)	24.58 (2.61)	0.87 (0.042)
Fixed seq.	28.48 (2.25)	0.88 (0.037)	30.05 (2.53)	0.93 (0.022)
Adaptive seq.	29.32 (1.72)	0.89 (0.031)	29.62 (2.08)	0.93 (0.020)
AEM	27.11 (2.17)	0.85 (0.048)	-	-

Table 5.1: Average (standard deviation) peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) over reconstructions of 32 images distorted by the nonlinear diffusion operator. The results are presented separately for Gaussian noise and impulse noise.

converting the images to grayscale and downsampling to size  $256^2$  to make the computations feasible. The dataset was also used to compute the mean vector and covariance matrix of the approximation error required by AEM.

**5. Results.** We start by examining the quantitative results. Tables 5.1 and 5.2 show the averaged PSNR and SSIM for reconstructions over a batch of 32 images with the NLD and CF

	CF (Gaussian)		CF (Impulse)	
	PSNR	SSIM	PSNR	SSIM
No correction	27.39 (2.38)	0.87 (0.042)	26.21 (2.32)	0.88 (0.038)
Fixed seq.	28.01 (2.36)	0.88 (0.042)	29.97 (2.43)	0.93 (0.025)
Adaptive seq.	28.18 (2.26)	0.88 (0.040)	30.46 (2.86)	0.93 (0.029)
AEM	26.42 (2.04)	0.85 (0.046)	-	-

Table 5.2: Average peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) over reconstructions of 32 images distorted by the curvature flow operator. The results are presented separately for Gaussian noise and impulse noise.

	Time in seconds			
	NLD (Gaussian)	NLD (Impulse)	CF (Gaussian)	CF (Impulse)
No correction	10.84 (2.04)	28.06 (5.57)	9.05 (1.26)	26.35 (4.70)
Fixed seq.	18.36 (4.56)	230.24 (71.47)	13.74 (3.02)	123.27 (32.47)
Adaptive seq.	153.75 (84.22)	373.76 (123.52)	71.58 (15.76)	301.29 (84.81)
AEM	91.51 (15.43)	-	237.48 (26.98)	-

Table 5.3: Average computation times (standard deviations) of different methods over 32 images.

models, respectively. The general trend is clear: using the fixed sequential correction method gives a clear improvement compared to not correcting the model at all. With NLD model and Gaussian noise the difference in PSNR is about 1.5 dB and 0.01 in SSIM. The difference with impulse noise is even more pronounced with about 5.5 dB in PSNR and 0.06 in SSIM. With NLD model and Gaussian noise the adaptive method gives a slight improvement over the fixed method. Interestingly, for impulse noise the effect is the opposite, a matter we discuss in Section 5.1. For the CF model the difference between fixed and adaptive methods is negligible. The performance of AEM is slightly lower than using no correction with both models.

We then take a qualitative look at the reconstructed images. Figure 5.1 shows reconstructions with different correction methods for the NLD model approximated with a LD model with Gaussian noise. Without any correction the edges in the image are oversharpened since the LD model assumes higher diffusivity over the edges. The adaptive method is able to recover a bit more details than the fixed method, for example in the earring. Qualitatively the reconstruction with AEM is quite close to the reconstruction with fixed approximation, and at some parts a bit more blurry. Figure 5.2 shows the same reconstructions for the CF model. Here the difference between fixed and adaptive methods is not so clear. The AEM reconstruction is clearly the worst with the eyes not properly recovered. Reconstructions for the NLD model and impulse noise are shown in Figure 5.3. Here, correcting the model greatly increases the ability to separate details in the image. The fixed correction creates some wavy artefacts near the edges. The adaptive correction takes care of the artefacts and the text on



Figure 5.1: Reconstructions for the nonlinear diffusion model approximated with linear diffusion model with 3% Gaussian noise. The numbers in parentheses indicate peak signal to noise ratio (dB) and structural similarity index measure.

the hat is more clearly visible. However, there is still some noise left in some parts of the image. We will discuss this phenomenon in Section 5.1. Figure 5.4 shows the reconstructions with CF model and impulse noise. Again, correcting the model increases the amount of details in the image. Again, the fixed correction introduces some artefacts at the edges while the adaptive correction takes care of the artefacts.

Finally, we investigate the convergence of the sequences. Figures 5.6 and 5.5 show the evolution of the data fidelity  $F(A(u), y)$  with respect to true model over sequence iterations. The behavior is as expected: the adaptive method converges to a lower value than the fixed method, experimentally confirming the analysis of Theorem 3.4, which stated that taking small enough steps in the sequence with adaptive approximation ends up decreasing the value of the objective function with respect to the true model. Furthermore, with the NLD model, the reconstructions with AEM obtain a slightly higher value of the data fidelity than with fixed approximation, while not using any correction gives the highest value. For the CF model the order of AEM and no correction is reversed. We can see from Table 5.3 that the fixed correction method offers a compromise between reconstruction quality and computational effort, compared to the adaptive method. The computation time is about one order of magnitude lower with the NLD model and Gaussian noise. The difference is less drastic with other models and noise types, due to slower convergence of the linear problem.



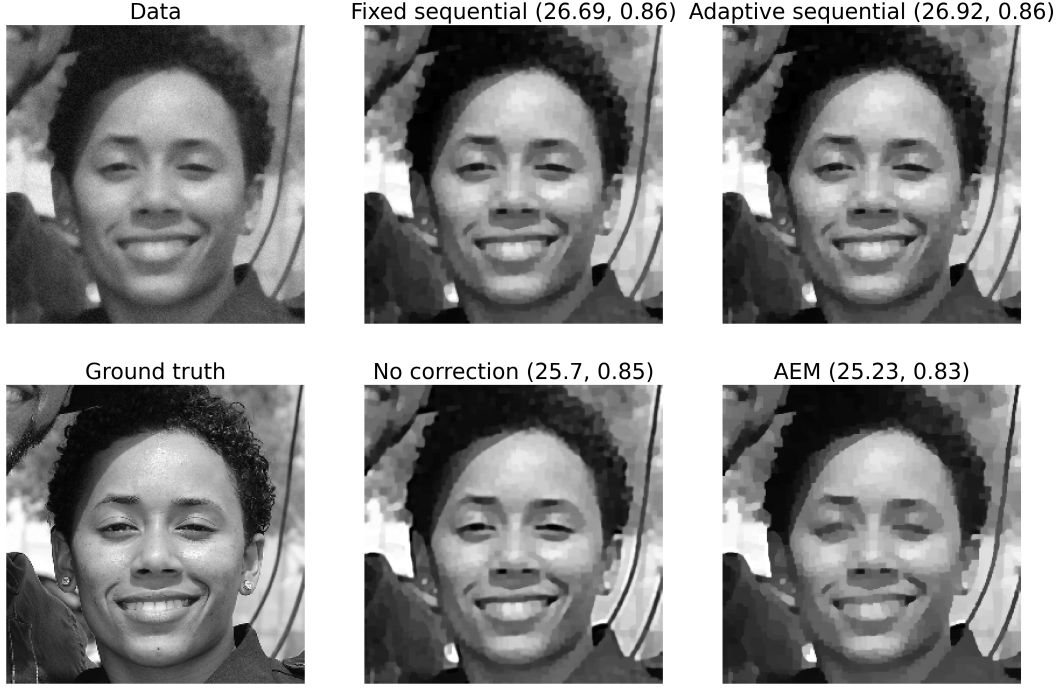


Figure 5.2: Reconstructions for the curvature flow model approximated with linear diffusion model with 3% Gaussian noise. The numbers in parentheses indicate peak signal to noise ratio (dB) and structural similarity index measure.

**5.1. Discussion of results.** With the NLD model and impulse noise, the fixed method gives quantitatively better results than the adaptive method. This is mainly due to the noise left in the reconstruction (see Figure 5.3). While increasing the regularization parameter could get rid of the noise, the quality of the image would deteriorate in other locations, resulting in oversmoothing. A possible fix could be to consider other kinds of data fidelity models, such as Cauchy noise model [31]. It penalises outliers even less than the  $L^1$  data fidelity which is based on the Laplace distribution. As we consider only convex data fidelity terms, we leave the Cauchy model to future studies as it is non-convex.

The computation time differences between fixed and adaptive methods are less pronounced for  $L^1$  data fidelity, especially with the NLD model. A possible reason could be the slower convergence of the fixed method which converges in 15–20 iterations while the adaptive method converges in about 10–15 iterations on average. Another reason for the relative small differences in computation time is the efficiency of the computation of the Jacobian that comes from the autograd library in PyTorch utilizing GPU parallelization.

As discussed earlier, AEM is based on the Gaussianity assumption of the approximation error. Figure 5.7 illustrates some of the pairwise distributions of the approximation error between the models used in this work. All in all, the distributions seem very non-Gaussian, with star-like shapes and outliers quite far from the center of the distribution. Even though

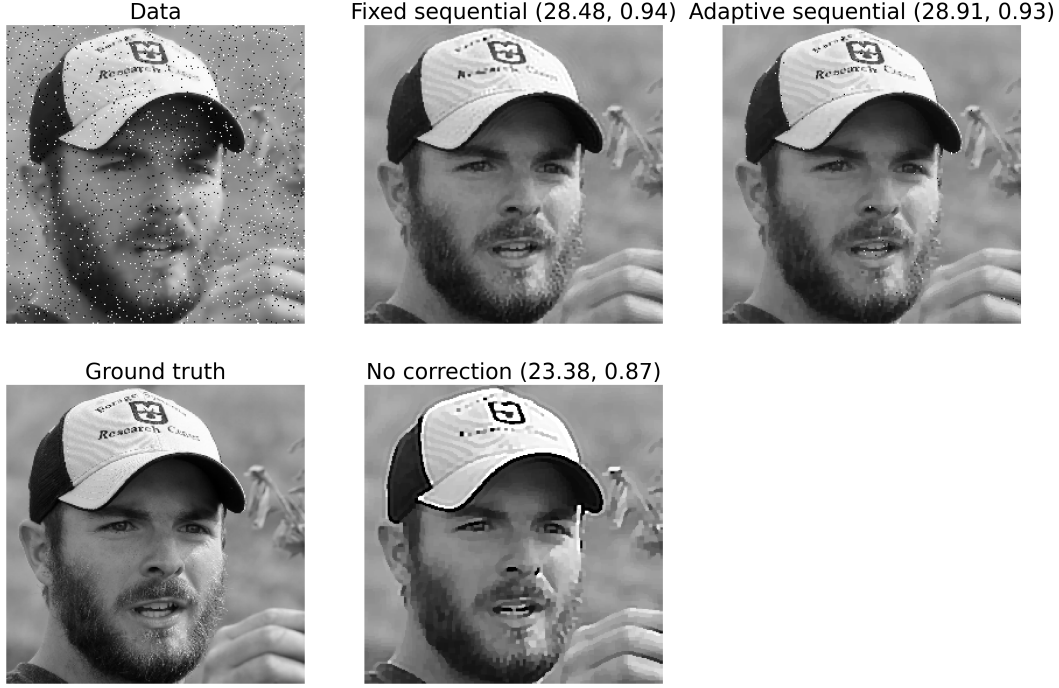


Figure 5.3: Reconstructions for the nonlinear diffusion model approximated with linear diffusion model with 4% salt and pepper noise. The numbers in parentheses indicate peak signal to noise ratio (dB) and structural similarity index measure.

the distribution is not Gaussian, the reconstructions with AEM with the NLD model achieve on average lower value of the data fidelity than not using any correction at all. It could be that the approximation error consists of multiple components of which some are Gaussian. AEM learns that part and leaves the rest of the error to be modeled as noise. We note that there exist also hierarchical constructions of the AEM, namely where the unknown is modeled as conditionally Gaussian (see e.g., [5]). Hierarchical modeling of Gaussian variables brings some flexibility and adaptiveness to model and can help in recovering non-Gaussian structures. We note that recently proposed neural network-based correction methods are also capable of correcting non-Gaussian errors but require training data to use [20, 33].

**6. Conclusion.** In this work we have proposed a strategy for correcting nonlinear models in the variational framework from a model correction perspective. We started from the observation that the conventional method for model correction that assumes a Gaussian distribution for the approximation error is not suitable for correcting non-Gaussian errors. These kind of errors arise especially when trying to correct nonlinear models with a linear approximation, as discussed in Section 2.1. The proposed strategy involves finding a linear approximation of the nonlinear model and solving the arising convex variational problem using the linear model. Updating the approximation error at the solution of the variational problem, the process is repeated until convergence. We investigated two different kinds of approximation,

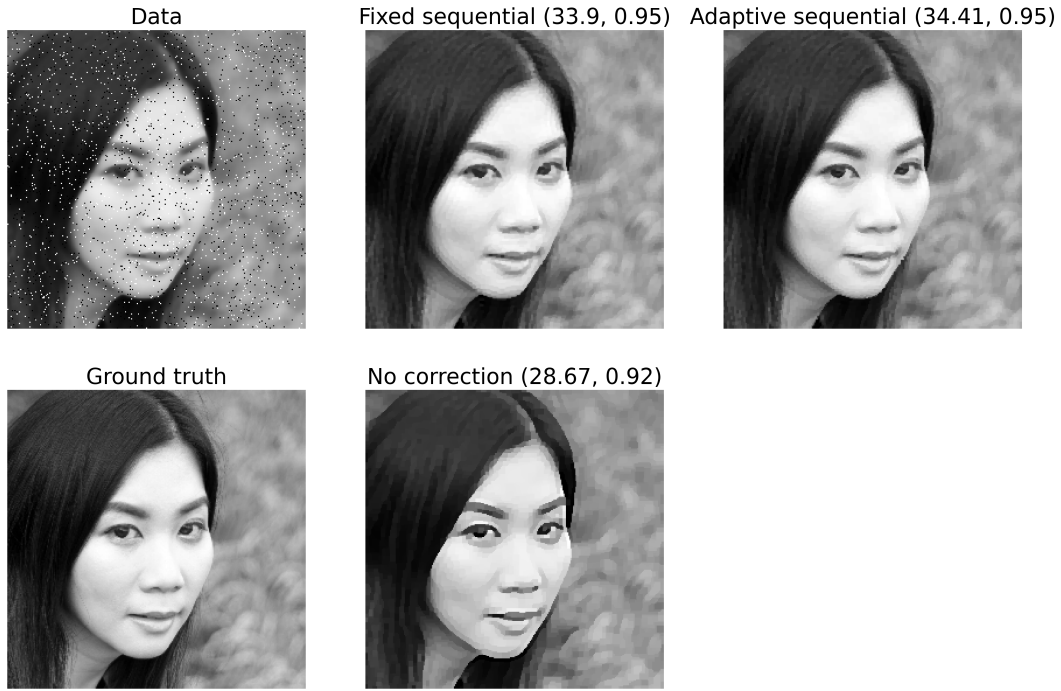


Figure 5.4: Reconstructions for the curvature flow model approximated with linear diffusion model with 4% salt and pepper noise. The numbers in parentheses indicate peak signal to noise ratio (dB) and structural similarity index measure.

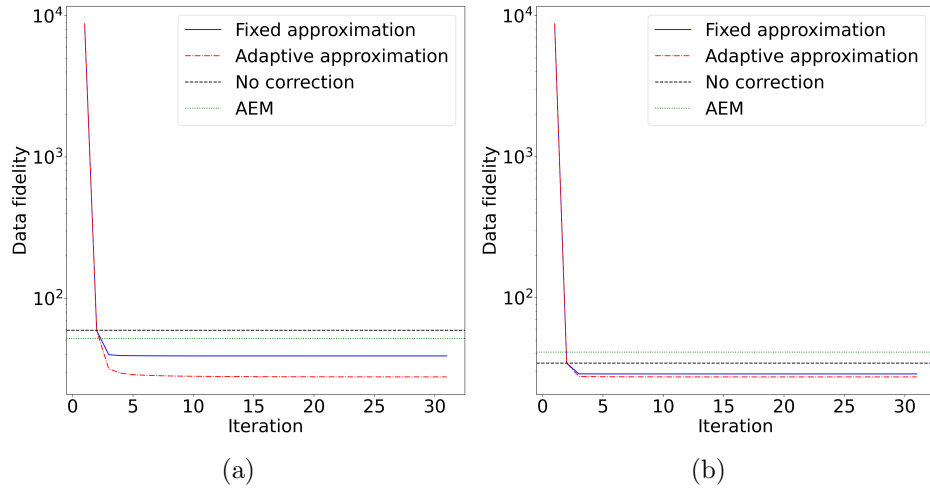


Figure 5.5: Average of the  $L_2$  data fidelity over reconstructions of 32 images with Gaussian noise for (a) nonlinear diffusion and (b) curvature flow models.

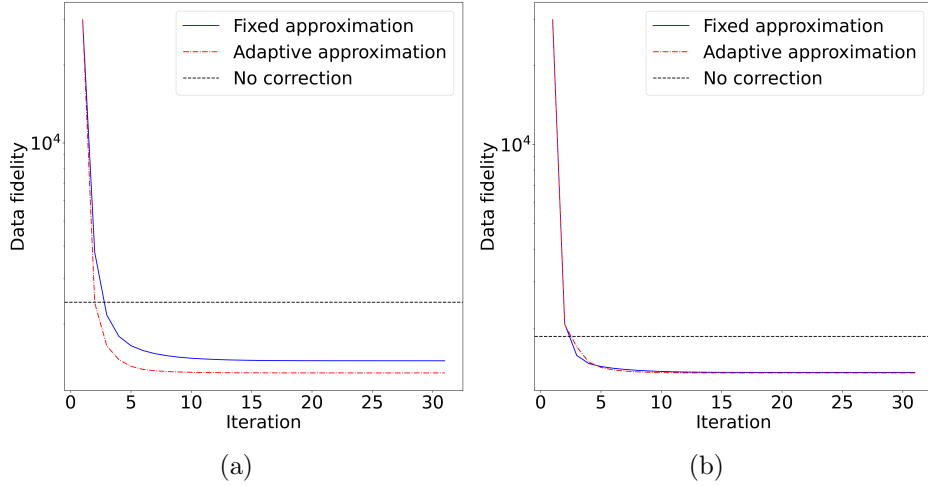


Figure 5.6: Average of the  $L_1$  data fidelity over reconstructions of 32 images with impulse noise for (a) nonlinear diffusion and (b) curvature flow models.

fixed and adaptive. The fixed approximation does not depend on the iteration number, making it computationally and conceptually simple to use, only requiring a few evaluations of the accurate nonlinear model. The sequence with the fixed approximation can be thought of as a fixed-point iteration, with simple conditions telling whether the sequence converges or not. Unfortunately, in practice it is rather difficult to tell if the conditions are fulfilled. Another possibility to ensure convergence is to terminate the sequence when the objective function with the exact operator can no longer be decreased. This involves evaluations of the correct model and might not be feasible if one evaluation is time consuming.

The adaptive approximation is different for every sequence iteration. We focused particularly on the case where the approximation is given as a first-order Taylor expansion centered at the current iterate. With certain smoothness assumptions for the model and objective function, we proved that the sequence with adaptive approximation always decreases the objective function with respect to the exact model. We also showed that with quadratic objective functions the adaptive sequence corresponds to the classic Gauss-Newton method.

The connection to Gauss-Newton methods could inspire further research to establish connections between model correction and nonlinear optimization. In many cases it is computationally prohibitive to compute the Jacobian of a nonlinear operator. Here, other approximations that satisfy the convergence criterion of Theorem 3.4 would be of further interest for future studies. Additionally, computationally cheap approximations of the Jacobian could be of interest, for instance by (learned) Quasi-Newton methods [34].

Finally, we will consider the application of the sequential approximation for the use with other nonlinear PDE based inverse problems [22, 29]. Here, we believe that the fixed approximation without the need to compute the Jacobian could be of great computational advantage.

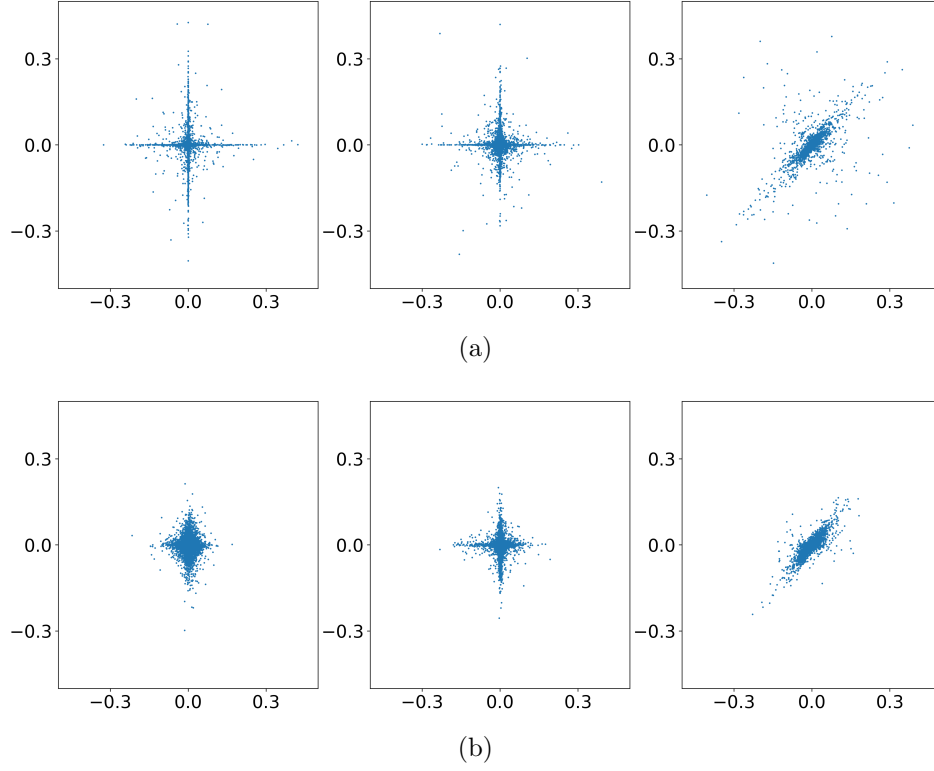


Figure 5.7: Scatterplots of some elements of the approximation error between (a) nonlinear and linear diffusion and (b) curvature flow and linear diffusion operators. These figures highlight non-Gaussianity in the approximation error and thus the need for a non-Gaussian correction.

**Acknowledgements.** Much of the theory was developed during Arttu Arjas’ visit to Marcelo Pereyra in Heriot-Watt University in Edinburgh. We also want to thank Simon Arridge for helpful discussions.

## REFERENCES

- [1] S. ARRIDGE AND A. HAUPTMANN, *Networks for nonlinear diffusion problems in imaging*, Journal of Mathematical Imaging and Vision, 62 (2019), pp. 471–487.
- [2] S. ARRIDGE, J. KAIPIO, V. KOLEHMAINEN, M. SCHWEIGER, E. SOMERSALO, T. TARVAINEN, AND M. VAUHKONEN, *Approximation errors and model reduction with an application in optical diffusion tomography*, Inverse problems, 22 (2006), p. 175.
- [3] S. ARRIDGE, P. MAASS, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Solving inverse problems using data-driven models*, Acta Numerica, 28 (2019), p. 1–174.
- [4] M. BENNING AND M. BURGER, *Modern regularization methods for inverse problems*, Acta Numerica, 27 (2018), pp. 1–111.
- [5] D. CALVETTI, A. COSMO, S. PEROTTO, AND E. SOMERSALO, *Bayesian mesh adaptation for estimating distributed parameters*, SIAM Journal on Scientific Computing, 42 (2020), pp. A3878–A3906.
- [6] V. CANDIANI, N. HYVÖNEN, J. P. KAIPIO, AND V. KOLEHMAINEN, *Approximation error method for*

- imaging the human head by electrical impedance tomography*, Inverse Problems, 37 (2021), p. 125008.
- [7] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.
  - [8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, JHU Press, 2013.
  - [9] R. C. GONZALEZ AND R. E. WOODS, *Digital Image Processing*, Pearson, 2008.
  - [10] W. W. HAGER, *Updating the inverse of a matrix*, SIAM Review, 31 (1989), pp. 221–239.
  - [11] N. HÄNNINEN, A. PULKKINEN, A. LEINO, AND T. TARVAINEN, *Application of diffusion approximation in quantitative photoacoustic tomography in the presence of low-scattering regions*, Journal of Quantitative Spectroscopy and Radiative Transfer, 250 (2020), p. 107065.
  - [12] A. HAUPTMANN, B. COX, F. LUCKA, N. HUYNH, M. BETCKE, P. BEARD, AND S. ARRIDGE, *Approximate k-space models and deep learning for fast photoacoustic reconstruction*, in International Workshop on Machine Learning for Medical Image Reconstruction, Springer, 2018, pp. 103–111.
  - [13] H. V. HENDERSON AND S. R. SEARLE, *On deriving the inverse of a sum of matrices*, Siam Review, 23 (1981), pp. 53–60.
  - [14] J. KAIPIO AND E. SOMERSALO, *Statistical and computational inverse problems*, vol. 160, Springer Science & Business Media, 2006.
  - [15] T. KARRAS, S. LAINE, AND T. AILA, *A style-based generator architecture for generative adversarial networks*, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 06 2019, pp. 4396–4405.
  - [16] J. KOPONEN, T. LÄHIVAARA, J. KAIPIO, AND M. VAUHKONEN, *Model reduction in acoustic inversion by artificial neural network*, The Journal of the Acoustical Society of America, 150 (2021), pp. 3435–3444.
  - [17] O. KOSHELEVA, *Babylonian method of computing the square root: Justifications based on fuzzy techniques and on computational complexity*, in NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society, 2009, pp. 1–6.
  - [18] K. LANGE, D. R. HUNTER, AND I. YANG, *Optimization transfer using surrogate objective functions*, Journal of Computational and Graphical Statistics, 9 (2000), pp. 1–20.
  - [19] B. LIU, Q. ZHANG, AND G. G. E. GIELEN, *A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems*, IEEE Transactions on Evolutionary Computation, 18 (2014), pp. 180–192.
  - [20] S. LUNZ, A. HAUPTMANN, T. TARVAINEN, C.-B. SCHÖNLIEB, AND S. ARRIDGE, *On learned operator correction in inverse problems*, SIAM Journal on Imaging Sciences, 14 (2021), pp. 92–127.
  - [21] F. MARGOTTI AND E. HAFEMANN, *Range-relaxed strategy applied to the levenberg–marquardt method with uniformly convex penalization term in banach spaces*, Inverse Problems, 38 (2022), p. 095001.
  - [22] M. MOZUMDER, A. HAUPTMANN, I. NISSILÄ, S. R. ARRIDGE, AND T. TARVAINEN, *A model-based iterative learning approach for diffuse optical tomography*, IEEE Transactions on Medical Imaging, 41 (2021), pp. 1289–1299.
  - [23] J. L. MUELLER AND S. SILTANEN, *Linear and Nonlinear Inverse Problems with Practical Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012.
  - [24] A. NISSINEN, V. P. KOLEHMAINEN, AND J. P. KAIPIO, *Compensation of modelling errors due to unknown domain boundary in electrical impedance tomography*, IEEE transactions on medical imaging, 30 (2010), pp. 231–242.
  - [25] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer New York, 2 ed., 2006.
  - [26] A. PASZKE, S. GROSS, F. MASSA, A. LERER, J. BRADBURY, G. CHANAN, T. KILLEEN, Z. LIN, N. GIMELSHEIN, L. ANTIGA, A. DESMAISON, A. KOPF, E. YANG, Z. DEVITO, M. RAISON, A. TEJANI, S. CHILAMKURTHY, B. STEINER, L. FANG, J. BAI, AND S. CHINTALA, *PyTorch: An imperative style, high-performance deep learning library*, in Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.
  - [27] P. PERONA AND J. MALIK, *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990), pp. 629–639.
  - [28] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.
  - [29] T. SAHLSTRÖM, A. PULKKINEN, J. TICK, J. LESKINEN, AND T. TARVAINEN, *Modeling of errors due to uncertainties in ultrasound sensor locations in photoacoustic tomography*, IEEE Transactions on Medical Imaging, 39 (2020), pp. 2140–2150.



- [30] M. SCHWEIGER, S. R. ARRIDGE, AND I. NISSLÄ, *Gauss-newton method for image reconstruction in diffuse optical tomography*, Physics in Medicine & Biology, 50 (2005), p. 2365.
- [31] F. SCIACCHITANO, Y. DONG, AND T. ZENG, *Variational approach for restoring blurred images with cauchy noise*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 1894–1922.
- [32] E. Y. SIDKY, J. H. JØRGENSEN, AND X. PAN, *Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm*, Physics in Medicine & Biology, 40 (2012), pp. 3065–3091.
- [33] D. SMYL, T. N. TALLMAN, J. A. BLACK, A. HAUPTMANN, AND D. LIU, *Learning and correcting non-gaussian model errors*, Journal of Computational Physics, 432 (2021), p. 110152.
- [34] D. SMYL, T. N. TALLMAN, D. LIU, AND A. HAUPTMANN, *An efficient quasi-newton method for nonlinear inverse problems via learned singular values*, IEEE Signal Processing Letters, 28 (2021), pp. 748–752.
- [35] T. TARVAINEN, V. KOLEHMAINEN, A. PULKKINEN, M. VAUHKONEN, M. SCHWEIGER, S. ARRIDGE, AND J. KAIPIO, *An approximation error approach for compensating for modelling errors between the radiative transfer equation and the diffusion approximation in diffuse optical tomography*, Inverse Problems, 26 (2009), p. 015005.
- [36] X. WANG, Z. TAN, N. SCHOLAND, V. ROELOFFS, AND M. UECKER, *Physics-based reconstruction methods for magnetic resonance imaging*, Philosophical Transactions of the Royal Society A, 379 (2021), p. 20200196.
- [37] Z. WANG, A. BOVIK, H. SHEIKH, AND E. SIMONCELLI, *Image quality assessment: from error visibility to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612.
- [38] J. WEICKERT, *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998.
- [39] L. YAN AND T. ZHOU, *An adaptive surrogate modeling based on deep neural networks for large-scale bayesian inverse problems*, Communications in Computational Physics, 28 (2020), pp. 2180–2205.