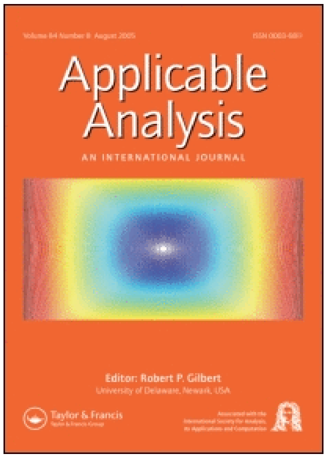

Applicable Analysis
An International Journal
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713454076

On regularization methods based on dynamic programming techniques

S. Kindermann [a]
[a] Institute for Industrial Mathematics, Johannes Kepler University. A-4040 Linz. Austria

To cite this Article: Kindermann, S. and Leitão, A. , 'On regularization methods based on dynamic programming techniques', Applicable Analysis, 86:5, 611 - 632
To link to this article: DOI: 10.1080/00036810701354953
URL: http://dx.doi.org/10.1080/00036810701354953

Applicable Analysis
Vol. 86, No. 5, May 2007, 611–632

On regularization methods based on dynamic programming techniques

S. KINDERMANN† and A. LEITÃO*‡

†Institute for Industrial Mathematics, Johannes Kepler University, A-4040 Linz, Austria
‡Department of Mathematics, Federal University of St. Catarina, 88040-900 Florianopolis, Brazil

Communicated by R. P. Gilbert

(*Received 30 September 2006; in final form 23 April 2007*)

In this article, we investigate the connection between *regularization theory* for inverse problems and *dynamic programming* theory. This is done by developing two new regularization methods, based on dynamic programming techniques. The aim of these methods is to obtain stable approximations to the solution of linear inverse ill-posed problems. We follow two different approaches and derive a continuous and a discrete regularization method. Regularization properties for both methods are proved as well as rates of convergence. A numerical benchmark problem concerning integral operators with convolution kernels is used to illustrate the theoretical results.

Keywords: Inverse problems; Regularization; Dynamic programming

AMS Classifications: 65J22; 49N45

1. Introduction

Our main goal is to establish a connection between *regularization theory* [7,14] for inverse problems and *dynamic programming* theory [2–6] for optimal control problems of linear quadratic type. This is done by developing two new regularization methods, based on dynamic programming results. The first one is a continuous regularization method, derived from the Hamilton–Jacobi equation and the Pontryagin maximum principle. The second is a discrete regularization method, derived from the Bellman optimality principle.

*Corresponding author. Email: aleitao@mtm.ufsc.br

Applicable Analysis ISSN 0003-6811 print/ISSN 1563-504X online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00036810701354953

In the sequel we describe the inverse problems we are concerned with. Let X, Y be Hilbert spaces. Consider the problem of finding $u \in X$ from the equation

$$Fu = y, \tag{1}$$

where $y \in Y$ represents the data and $F : X \to Y$ is a linear ill-posed operator. Since the operator F is ill-posed, the solution u does not depend in a stable way on the right hand side y and regularization techniques have to be used in order to obtain a stable solution. Continuous and discrete regularization methods have been quite well studied in the last two decades and one can find relevant information in [7–10,14,15] and in the references therein. The aim of these methods is to obtain stable approximations to the solution of the inverse problem (1).

Next, we give a brief description of the optimal control problems (continuous and discrete) that will serve as starting point for developing the regularization methods in this article. These problems are mainly characterized by possessing a linear dynamic and a quadratic objective function.

Our first (continuous) approach is based on the the following constrained optimization problem:

$$\begin{cases} \text{Minimize } J(x, w) := \displaystyle\int_0^T \langle x(t), Lx(t)\rangle + \langle w(t), Mw(t)\rangle \mathrm{d}t \\ \text{s.t.} \\ x' = Ax + Bw, \quad t \geq 0, \ x(0) = x_0, \end{cases} \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the system trajectory, $w(t) \in \mathbb{R}^m$, $t \geq 0$ is the control variable, A, $L \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $M \in \mathbb{R}^{n,n}$ are given matrices and $x_0 \in \mathbb{R}^n$ is the initial condition. The goal of the control problem is to find a pair of functions (x, w), minimizing the quadratic objective function J and satisfying the constraint imposed by the linear dynamic system – such pairs are called *admissible processes*. In this article, we adapt a solution technique for this problem (dynamic programming) in order to derive a continuous regularization method for the inverse problem (1).

Our second (discrete) approach, has as starting point the discrete optimal control problem

$$\begin{cases} \text{Minimize } J(x, w) := \langle x_N,\ Sx_N\rangle + \displaystyle\sum_{k=0}^{N-1} \langle x_k, Lx_k\rangle + \langle w_k, Mw_k\rangle \\ \text{s.t.} \\ x_{k+1} = Ax_k + Bw_k, \quad k = 0, \ldots, N-1, \ x_0 \in \mathbb{R}^n. \end{cases} \tag{3}$$

The matrices A, B, L, M have the same meaning as in problem (2) and $S \in \mathbb{R}^{n,n}$ is positive definite. Notice that the final time T in (2) is substituted by the number of discrete steps $N \in \mathbb{N}$ in (3). Again, using the dynamic programming technique, we are able to derive a discrete regularization method for the inverse problem (1). In this discrete framework, the dynamic programming approach consists basically of the Bellman optimality principle and the dynamic programming equation.

To the best of our knowledge, dynamic programming techniques have only been applied to solve particular inverse problems so far. In [11] the inverse problem of identifying the initial condition in a semilinear parabolic equation is considered.

In [12] the same authors consider a parameter identification problem for identification of systems of distributed parameters. In this article, however, dynamic programming methods allow us to formulate regularization methods in an abstract functional analytical framework for general inverse problems.

The article is outlined as follows: In section 2, we derive both regularization methods (continuous and discrete). In section 3, we analyze regularization properties of the proposed methods. Rates of convergence are derived under abstract source conditions and an *a priori* parameter choice yielding optimal order convergence rates is provided. Furthermore, for the discrete regularization method, we characterize the filter functions (for the regularization operator) in terms of Chebyshev polynomials. In section 4, numerical realizations of our regularization methods are presented. We use our methods to solve an integral equation of the first kind and compare the obtained performances with the Landweber iteration and with the CG-method.

2. Derivation of the regularization methods

2.1. *A continuous approach*

We start this section defining an optimal control problem related with the linear inverse problem (1). Let $u_0 \in X$ be any approximation for the minimum norm solution $u^\dagger \in X$ of (1). We aim to find a function $u : [0, T] \to X$ such that, $u(0) = u_0$ and

$$\|Fu(T) - y\| \approx \|Fu^\dagger - y\|. \tag{4}$$

In the control literature, the function u is called *trajectory* (or *state*) and its evolution is described by a dynamical system. For simplicity, we choose a linear evolution model, i.e. $u' = Au(t) + Bv(t)$, $t \geq 0$, where $A, B : X \to X$ are linear operators and $v : [0, T] \to X$ is the *control* of the system (compare with the classical problem in (2)). Since our main concern is to satisfy the property described in (4), it is enough for our purpose to consider a simpler dynamic, which does not depend on the state u, but only on the control v. This justifies the choice of the dynamic: $u' = v$, $t \geq 0$. In this case, the control v corresponds to a *velocity function*.

The next step is to choose the objective function for our control problem. Recalling the formulation of the linear quadratic control problem in (2) and also the goals described in (4), the objective function has to be related to the minimization of both the residual norm and the velocity norm along the trajectories, i.e.,

$$J(u, v) := \frac{1}{2}\int_0^T \|Fu(t) - y\|^2 + \|v(t)\|^2 \, \mathrm{d}t.$$

Putting all together we obtain the following abstract optimal control problem in Hilbert spaces:

$$\begin{cases} \text{Minimize } J(u, v) = \dfrac{1}{2}\displaystyle\int_0^T \|Fu(t) - y\|^2 + \|v(t)\|^2 \, \mathrm{d}t \\ \text{s.t.} \\ u' = v, \quad t \geq 0, \; u(0) = u_0, \end{cases} \tag{5}$$

where the (fixed but arbitrary) final time $T > 0$ will play the role of the regularization parameter. The functions $u, v: [0, T] \to X$ correspond respectively to the trajectory and the control of the system, and the pairs (u, v) are called *processes*.

Next we define the residual function $\varepsilon(t) := Fu(t) - y$ associated to a given trajectory u. Notice that this residual function evolves according to the dynamic

$$\varepsilon' = Fu'(t) = Fv(t), \quad t \geq 0.$$

With this notation, problem (5) can be rewritten in the following form

$$\begin{cases} \text{Minimize } J(\varepsilon, v) = \dfrac{1}{2}\displaystyle\int_0^T \|\varepsilon(t)\|^2 + \|v(t)\|^2 \, \mathrm{d}t \\ \text{s.t.} \\ \varepsilon' = Fv, \quad t \geq 0, \ \varepsilon(0) = Fu_0 - y. \end{cases} \tag{6}$$

The next result states a parallel between solvability of the optimal control problem (5) and the auxiliary problem (6).

Proposition 2.1 *If $(\bar{u}, \bar{v})$ is an optimal process for problem* (5), *then the process $(\bar{\varepsilon}, \bar{v})$, with $\bar{\varepsilon} := F\bar{u} - y$, will be an optimal process for problem* (6). *Conversely, if $(\bar{\varepsilon}, \bar{v})$ is an optimal process for problem* (6), *with $\varepsilon(0) = Fu_0 - y$, for some $u_0 \in X$, then the process $(\bar{u}, \bar{v})$ is an optimal process for problem* (5).

In the sequel, we derive the dynamic programming approach for the optimal control problem in (6). We start by introducing the first Hamilton function. This is the function $H : \mathbb{R} \times Y^2 \times X \to \mathbb{R}$ given by

$$H(t, \varepsilon, \lambda, v) := \langle \lambda, Fv \rangle + \frac{1}{2}[\langle \varepsilon, \varepsilon \rangle + \langle v, v \rangle].$$

Notice that the variable λ plays the role of a Lagrange multiplier in the above definition. According to the Pontryagin's maximum principle, the Hamilton function furnishes a necessary condition of optimality for problem (6). Furthermore, since this function (in this particular case) is convex in the control variable, this optimality condition also happens to be sufficient. Recalling the maximum principle, along an optimal trajectory we must have

$$0 = \frac{\partial H}{\partial v}(t, \varepsilon(t), \lambda(t), v(t)) = F^*\lambda(t) + v(t). \tag{7}$$

This means that the optimal control $\bar{v}$ can be obtained directly from the Lagrange multiplier $\lambda : [0, T] \to Y$, by the formula

$$\bar{v}(t) = -F^*\lambda(t), \quad \forall t.$$

Therefore, the key task is actually the evaluation of the Lagrange multiplier. This leads us to the Hamilton–Jacobi equation. Substituting the above expression for $\bar{v}$ in (7), we can define the second Hamilton function $\mathcal{H} : \mathbb{R} \times Y^2 \to \mathbb{R}$

$$\mathcal{H}(t, \varepsilon, \lambda) := \min_{v \in X}\{H(t, \varepsilon, \lambda, v)\} = \frac{1}{2}\langle \varepsilon, \varepsilon \rangle - \frac{1}{2}\langle \lambda, FF^*\lambda \rangle.$$

Now, let $V : [0, T] \times X \to \mathbb{R}$ be the value function for problem (6), i.e.,

$$V(t, \xi) := \min\left\{\frac{1}{2}\int_t^T \|\varepsilon(s)\|^2 + \|v(s)\|^2 \, \mathrm{d}s | (\varepsilon, v) \text{ admissible process} \right.$$
$$\left. \text{for problem (6) with initial condition } \varepsilon(t) = \xi\right\}. \tag{8}$$

The interest in the value function follows from the fact that this function is related to the Lagrange multiplier λ by the formula: $\lambda(t) = \partial V/\partial\varepsilon(t, \bar{\varepsilon})$, where $\bar{\varepsilon}$ is an optimal trajectory.

From the control theory, we know that the value function is a solution of the Hamilton–Jacobi equation

$$\frac{\partial V}{\partial t}(t, \varepsilon) + \mathcal{H}\left(t, \varepsilon, \frac{\partial V}{\partial \varepsilon}(t, \varepsilon)\right) = 0. \tag{9}$$

Now, making the ansatz: $V(t, \varepsilon) = (1/2)\langle\varepsilon, Q(t)\varepsilon\rangle$, with $Q : [0, T] \to \mathbb{R}$, we are able to rewrite (9) in the form

$$\langle\varepsilon, Q'(t)\varepsilon\rangle + \langle\varepsilon, \varepsilon\rangle - \langle Q(t)\varepsilon, FF^*Q(t)\varepsilon\rangle = 0.$$

Since this equation must hold for all $\varepsilon \in X$, the function Q can be obtained by solving the Riccati equation

$$Q'(t) = -I + Q(t)FF^*Q(t). \tag{10}$$

Notice that the cost of all admissible processes for an initial condition of the type (T, ε) is zero. Therefore, we have to consider the Riccati equation (10) with the final condition

$$Q(T) = 0. \tag{11}$$

Once we have solved the initial value problems (10), (11), the Lagrange multiplier is given by $\lambda(t) = Q(t)\bar{\varepsilon}(t)$ and the optimal control is obtained by the formula $\bar{v}(t) = -F^*Q(t)\bar{\varepsilon}(t)$. Therefore, the optimal trajectory of problem (5) is defined via

$$\bar{u}' = -F^*Q(t)[F\bar{u}(t) - y], \quad \bar{u}(0) = u_0. \tag{12}$$

We use the optimal trajectory defined by the initial value problem (12) in order to define a family of reconstruction operators $R_T : X \to X$, $T \in \mathbb{R}^+$,

$$R_T(y) := \bar{u}(T) = u_0 - \int_0^T F^*Q(t)[F\bar{u}(t) - y]\mathrm{d}t. \tag{13}$$

We shall return to the operators $\{R_T\}$ in section 3 and prove that the family of operators defined in (13) is a regularization method for (1) [7].

2.2. *A discrete approach*

In this section, we use the optimal control problem (3) as starting point to derive a discrete reconstruction method for the inverse problem in (1). Again, let $u_0 \in X$ be a given approximation for the minimum norm solution $u^\dagger \in X$ of (1) and $N \in \mathbb{N}$. Analogously as we did in the previous section, we aim to find a sequence $\{u_k\}_{k=1}^N$ in X, starting from $u_0 = u_0$, such that

$$\|Fu_N - y\| \approx \|Fu^\dagger - y\|. \tag{14}$$

As in the previous section, we have now a discrete trajectory, represented by the sequence u_k, which evolution is described by the discrete dynamic

$$u_{k+1} = Au_k + Bv_k, \quad k = 0, 1, \ldots$$

where the operators A and B are defined as before and $\{v_k\}_{k=0}^{N-1}$, is the control of the system [compare with (3)]. As in the continuous case, we shall consider a simpler dynamic: $u_{k+1} = u_k + v_k$, $k = 0, 1, \ldots$ (i.e., $A = B = I$). To simplify the notation, we represent the processes $(u_k, v_k)_{k=1}^N$ by (u, v).

The objective function is chosen similarly as in the continuous case:

$$J(u, v) := \frac{1}{2}\langle Fu_N - y, S(Fu_N - y)\rangle + \frac{1}{2}\sum_{k=0}^{N-1} \|Fu_k - y\|^2 + \|v_k\|^2,$$

with some positive operator $S : Y \to Y$. Putting all together, we obtain the following abstract optimal control problem in Hilbert spaces:

$$\begin{cases} \text{Minimize } J(u, v) = \frac{1}{2}\langle Fu_N - y, S(Fu_N - y)\rangle + \frac{1}{2}\sum_{k=0}^{N-1}\|Fu_k - y\|^2 + \|v_k\|^2 \\ \text{s.t.} \\ u_{k+1} = u_k + v_k, \quad k = 0, 1, \ldots, u_0 \in X \end{cases} \tag{15}$$

where the (fixed but arbitrary) number of discrete steps $N \in \mathbb{N}$ will play the role of the regularization parameter.

As in the continuous approach, we define the residual sequence $\varepsilon_k := Fu_k - y$, associated to a given trajectory u. Notice that

$$\varepsilon_{k+1} = Fu_{k+1} - y = \varepsilon_k + Fv_k, \quad k = 0, 1, \ldots$$

With this notation, problem (15) can be rewritten in the form

$$\begin{cases} \text{Minimize } J(\varepsilon, v) = \frac{1}{2}\langle \varepsilon_N, S\varepsilon_N\rangle + \frac{1}{2}\sum_{k=0}^{N-1}\|\varepsilon_k\|^2 + \|v_k\|^2 \\ \text{s.t.} \\ \varepsilon_{k+1} = \varepsilon_k + Fv_k, \quad k = 0, 1, \ldots, \varepsilon_0 = Fu_0 - y. \end{cases} \tag{16}$$

Notice that Proposition 2.1 holds also for the discrete case, i.e. if $(\bar{u}, \bar{v})$ is an optimal process for problem (15), then the process $(\bar{\varepsilon}, \bar{v})$, with $\bar{\varepsilon}_k := F\bar{u}_k - y$, will be an optimal process for problem (16) and vice versa, as one can easily check.

In the sequel, we derive the dynamic programming approach for the optimal control problem in (16). We start by introducing the value function (or Lyapunov function) $V : \mathbb{R} \times Y \to \mathbb{R}$,

$$V(k, \xi) := \min\left\{J_k(\varepsilon, v) | (\varepsilon, v) \in Z_k(\xi) \times X^{N-k}\right\},$$

where

$$J_k(\varepsilon, v) := \frac{1}{2}\left[\langle \varepsilon_N, S\varepsilon_N \rangle + \sum_{j=k}^{N-1} \|\varepsilon_j\|^2 + \|v_j\|^2\right]$$

and

$$Z_k(\xi) := \left\{\varepsilon \in Y^{N-k+1} | \varepsilon_k = \xi, \varepsilon_{j+1} = \varepsilon_j + Fv_j,\ j = k, \ldots, N-1\right\}.$$

[Compare with the definition in (8)]. The Bellman principle for this discrete problem reads

$$V(k, \xi) = \min\left\{V(k+1, \xi + Fv) + \frac{1}{2}(\langle \xi, \xi \rangle + \langle v, v \rangle) | v \in X\right\}. \tag{17}$$

The optimality equation (17) is the discrete counterpart of the Hamilton–Jacobi equation (9). Notice that the value function also satisfies the boundary condition: $V(N, \xi) = (1/2)\langle \xi, S\xi \rangle$.

As in the continuous case, the optimality equation have to be solved backwards in time $(k = N-1, \ldots, 1)$ recursively.

For $k = N-1$, we have

$$V(N-1, \xi) = \min\left\{\frac{1}{2}(\langle \xi + Fv, S(\xi + Fv) \rangle + \langle \xi, \xi \rangle + \langle v, v \rangle) | v \in X\right\}. \tag{18}$$

A necessary and sufficient condition for u_{N-1} to be a minimum of (18) is given by $v + F^*S(\xi + Fv) = 0$. Solving this equation for v, we obtain

$$\bar{v}_{N-1} := -(F^*SF + I)^{-1}F^*S\xi.$$

In order to obtain the optimal control recursively, we evaluate the matrices

$$\begin{aligned} &S_N := S; \\ &\text{for } k = N-1, \ldots, 0 \text{ evaluate} \\ &\quad R_k := (F^*S_{k+1}F + I)^{-1}F^*S_{k+1}; \\ &\quad S_k := (I - FR_k)^*S_{k+1}(I - FR_k) + R_k^*R_k + I; \end{aligned} \tag{19}$$

once the matrices R_k and S_k are known, we obtain the optimal control recursively, using the algorithm:

$$\begin{aligned} &\varepsilon_0 := Fu_0 - y;\\ &\text{for } k = 0, \ldots, N-1, \text{ evaluate}\\ &\qquad \bar{v}_k := -R_k \bar{\varepsilon}_k;\\ &\qquad \bar{u}_{k+1} := u_k + \bar{v}_k;\\ &\qquad \bar{\varepsilon}_{k+1} := \bar{\varepsilon}_k + F\bar{v}_k; \end{aligned} \tag{20}$$

to obtain the optimal control $\bar{v} = (\bar{v}_0, \ldots, \bar{v}_{N-1})$, the optimal trajectory for problem (16) $\bar{\varepsilon} = (\bar{\varepsilon}_0, \ldots, \bar{\varepsilon}_N)$, and the optimal trajectory for problem (15) $\bar{u} = (\bar{u}_0, \ldots, \bar{u}_N)$. Furthermore, the optimal cost is given by $V(0, \varepsilon_0) = (1/2)\langle \varepsilon_0, S_0 \varepsilon_0 \rangle$.

3. Regularization properties

3.1. *Regularization in the continuous case*

In this section, we investigate the regularization properties of the operator R_T introduced in (13). Consider the Riccati equation (10) for the operator Q: we may express the operator $Q(t)$ via the spectral family $\{F_\lambda\}$ of FF^* [7]. Hence, we make the ansatz

$$Q(t) = \int q(t, \lambda) \mathrm{d}F_\lambda.$$

Assuming that $q(t, \lambda)$ is C^1 we may find from (10) together with the boundary condition at $t = T$ that

$$\int \left(\frac{\mathrm{d}}{\mathrm{d}t} q(t, \lambda) + 1 - q(t, \lambda)^2 \lambda \right) \mathrm{d}F_\lambda = 0, \quad q(T, \lambda) = 0.$$

Hence, we obtain an ordinary differential equation for q:

$$\frac{\mathrm{d}}{\mathrm{d}t} q(t, \lambda) = -1 + \lambda q(t, \lambda)^2 \tag{21}$$

The solution to these equations is given by

$$q(t, \lambda) = -\frac{1}{\sqrt{\lambda}} \tanh\left(\sqrt{\lambda}(t - T)\right) = \frac{1}{\sqrt{\lambda}} \tanh\left(\sqrt{\lambda}(T - t)\right). \tag{22}$$

If $t < T$, then $Q(t)$ is nonsingular, since $\lim_{x \to 0}(\tanh(xa)/x) = a$ and $(\tanh(ax)/x)$ is monotonically decreasing for $x > 0$. Hence the spectrum of $Q(t)$ is contained in the interval $[(\tanh((T-t)\|F\|)/\|F\|, (T-t)]$. Now consider the evolution equation (12): The operator $Q(t)$ can be expressed as $Q(t) = q(t, FF^*)$; by usual spectral theoretic properties [7] it holds that

$$F^*q(t, FF^*) = q(t, F^*F)F^*.$$

Hence we obtain the problem

$$u'(t) = -q(t, F^*F)(F^*Fu(t) - F^*y) \tag{23}$$

$$u(0) = u_0. \tag{24}$$

We may again use an ansatz via spectral calculus: if we set

$$u(t) = \int g(t, \lambda)\mathrm{d}E_\lambda F^*y,$$

where E_λ is the spectral family of F^*F, we derive an ordinary differential equation for g. Similar as above, we can express the solution to (23, 24) in the form

$$u(t) = \int \frac{1 - (\cosh(\sqrt{\lambda}(T-t))/(\cosh(\sqrt{\lambda}T))}{\lambda}\mathrm{d}E_\lambda F^*y + \int \frac{\cosh(\sqrt{\lambda}(T-t))}{\cosh(\sqrt{\lambda}T)}\mathrm{d}E_\lambda u_0. \tag{25}$$

Setting $t = T$ we find an approximation of the solution

$$u_T := u(T) = \int \frac{1 - [1/(\cosh(\sqrt{\lambda}T))]}{\lambda}\mathrm{d}E_\lambda F^*y + \int \frac{1}{\cosh(\sqrt{\lambda}T)}\mathrm{d}E_\lambda u_0. \tag{26}$$

Note the similarity to Showalter's methods [7], where the term $\exp(\lambda T)$ instead of $\cosh(\sqrt{\lambda}T)$ appears.

Theorem 3.1 *The operator R_T in (13) is a regularization operator with qualification $\mu_0 = \infty$:*

If the data are exact, $y = Fu^\dagger$ and $u^\dagger$ satisfies a source condition for some $\mu > 0$

$$\exists \omega \in X:\ u^\dagger = (F^*F)^\mu \omega, \tag{27}$$

we have the estimate

$$\|u_T - u^\dagger\| \le C_\mu T^{-2\mu}$$

If the data are contaminated with noise, $\|y - y_\delta\| \le \delta$ and $y = Fu^\dagger$ with $u^\dagger$ as in (27), then we have

$$\|u_{T,\delta} - u^\dagger\| \le C_\mu T^{-2\mu} + \delta T.$$

In particular, the a-priori parameter choice $T \sim \delta^{(-1/(2\mu+1))}$ *yields the optimal order convergence rate*

$$\|u_{T,\delta} - u^\dagger\| \sim \delta^{(2/(2\mu+1))}.$$

Proof For simplicity we set $u_0 = 0$, the generalization to the inhomogeneous case is obvious. (26) gives an expression of the regularization operator in terms of a filter function:

$$R_T = \int f(T, \lambda) \mathrm{d}E_\lambda F^* y$$

with

$$f(T, \lambda) = \lambda^{-1}\left(1 - \frac{1}{\cosh(\sqrt{\lambda}T)}\right).$$

According to [7] we have to show that the filter function $f(T, \lambda)$ satisfies the properties (regarding $1/T$ as regularization parameter).

(1) for T fixed, $f(T, .)$ is continuous;
(2) there exists a constant C such that for all $\lambda > 0$

$$|\lambda f(T, \lambda)| \leq C;$$

(3)

$$\lim_{T\to\infty} f_T(\lambda) = \lambda^{-1}, \quad \forall \lambda \in (0, \|F^*F\|].$$

(1) is clear since $\lim_{\lambda\to 0} f(T, \lambda) = (T^2/2)$ the function can be extended continuously to $\lambda = 0$
(2) holds with $C = 1$ since $0 \leq 1/(\cosh(\sqrt{\lambda}(T))) \leq 1$.
(3) is obviously the case since $\lim\limits_{\delta\to\infty} \cosh(s) = \infty$.

We have to show that the qualification $\mu_0 = \infty$: this needs an estimate $w_\mu(T)$ such that

$$\lambda^\mu |(1 - \lambda f(T, \lambda))| \leq w_\mu(T).$$

It holds that

$$\lambda^\mu |(1 - \lambda f(T, \lambda))| = \frac{\lambda^\mu}{\cosh(\sqrt{\lambda}T)} \leq 2\frac{\lambda^\mu}{\exp(\sqrt{\lambda}T)} \leq 2(2\mu)^{2\mu} \exp(-2\mu) T^{-2\mu}.$$

Hence, for all $\mu > 0$, $w_\mu(T) \sim C_\mu T^{-2\mu}$ holds.

On the other hand, we see that $f(t, \lambda)$ is monotonically decreasing. Hence, it takes the maximum value at $\lambda = 0$:

$$\sup_{\lambda > 0} |f(t, \lambda)| \leq \frac{1}{2} T^2.$$

Using the results in [7] it follows immediately that with $\frac{1}{T^2} = \alpha$ we have a regularization operator of optimal order. ■

If we compare the dynamic programming approach with the Showalter method, they are quite similar with $T_{dyn}^2 \sim T_{Sw}$. Hence, to obtain the same order of convergence we only need $\sqrt{T_{Sw}}$ of the time for the Showalter method.

3.2. *Regularization in the discrete case*

The dynamic programming principle allows us to find a sequence of approximate solutions $\{u_k\}$ which is a minimizer to a certain functional.

From regularization theory, we are motivated to choose a functional which includes the norm of the residuals $\|Fu_k - y\|$. Since in general this will not necessarily yield a regularization, we include an additional term involving $u_{k+1} - u_k$. Now analogous to the continuous case we want to minimize the functional

$$J\big(\{u_k\}_{k=0}^N\big) := \sum_{j=0}^{N} \|Fu_k - y\|^2 + \sum_{i=0}^{N-1} \|u_{k+1} - u_k\|^2 \tag{28}$$

with respect to all sequences $\{u_k\}_{k=0}^N$ satisfying $u_0 = 0$. The reason for choosing the norm of the residuals is clear, since we want to find an (approximate) solution to the equation $Fu = y$. The second term is important to obtain a regularization method, since it controls the size of the steplength between two successive iterations.

At first sight it is not at all obvious that there is a constructive method for minimizing (28) with respect to all sequences $\{u_k\}_{k=0}^N$. However, we show that the minimization problem can be treated within the framework of subsection 2.2.

Define ε_k as the k-th residual: $\epsilon_k := Fu_k - y$, $k = 0, \ldots, N$, where u_k is the solution we compute at the k-th iteration step. The control is defined as $v_k = u_{k+1} - u_k$, $k = 0, \ldots, N-1$. As initial starting value we set $u_0 = 0$. Hence we obtain the k-th iterate from the control variables by

$$u_k = \sum_{j=0}^{k-1} v_j. \tag{29}$$

From these definitions we obtain the following condition, which is trivially satisfied, when v_k and ϵ_k are defined in this way:

$$\epsilon_{k+1} = \epsilon_k + Fv_k. \tag{30}$$

Using the above notations, the minimization of (28), with initial condition $u_0 = 0$, is equivalent to the optimization problem in (16).

We now can use the results of section 2.2 with $S = Q = R = A = I$, $B = F$. The dynamic programming principle yields the iteration procedure

$$S_N := I \tag{31}$$

$$K_k := (F^* S_{k+1} F + I)^{-1} F^* S_{k+1}, \quad k = N-1, \ldots, 0 \tag{32}$$

$$S_k := (I - FK_k)^* S_{k+1} (I - FK_k) + K_k^* K_k + I, \quad k = N-1, \ldots, 0 \tag{33}$$

If K_k, S_k are computed, we obtain the control v_k and the error ϵ_k from

$$\epsilon_0 := -y \tag{34}$$

$$v_k = -K_k\epsilon_k, \quad k = 0, \dots, N-1 \tag{35}$$

$$\epsilon_{k+1} = \epsilon_k + Fv_k = (I - FK_k)\epsilon_k. \tag{36}$$

The iterate u_N, which represents an approximation to the solution, can be calculated from (29).

Now we want to consider the mapping $y \to u_N$ as an iterative regularization operator where N acts as regularization parameter. This mapping can be represented by filter functions g_N using spectral theory, similar to the continuous case. The following lemma serves as preparation for this purpose. Let E_λ, F_λ be the spectral families of F^*F, FF^*.

Lemma 3.2 *If S_{k+1} has a representation as $S_{k+1} = \int f_{k+1}(\lambda)\mathrm{d}F_\lambda$, with a continuous positive function f_{k+1}, then so has $S_k = \int f_k(\lambda)\mathrm{d}F_\lambda$ and the following recursion formula holds*:

$$f_k(\lambda) = \frac{f_{k+1}(\lambda)(\lambda+1)+1}{f_{k+1}(\lambda)\lambda+1} = 1 + \frac{f_{k+1}}{f_{k+1}(\lambda)\lambda+1}. \tag{37}$$

Proof We use the identity $F^*f(FF^*) = f(F^*F)F^*$ (see [7], (2.43)), which holds for any piecewise continuous function f. Since f_{k+1} is positive, the inverse $(f_{k+1}(\lambda)\lambda+1)^{-1}$ exists, and

$$K_k = \int (f_{k+1}(\lambda)\lambda+1)^{-1} f_{k+1}(\lambda)\mathrm{d}E_\lambda F^*.$$

From the identity above and some basic algebraic manipulation we obtain

$$\begin{aligned} S_k &= \int (f_{k+1}(\lambda)\lambda+1)^{-2} f_{k+1} + (f_{k+1}(\lambda)\lambda+1)^{-2} f_{k+1}(\lambda)^2\lambda + 1\,\mathrm{d}F_\lambda \\ &= \int \frac{(f_{k+1}(\lambda)(\lambda+1)+1)}{(f_{k+1}(\lambda)\lambda+1)}\mathrm{d}F_\lambda = \int 1 + \frac{f_{k+1}}{f_{k+1}(\lambda)\lambda+1}\mathrm{d}F_\lambda. \end{aligned}$$

■

By definition we have $S_N = I$, f_N obviously satisfies the hypothesis of the theorem with $f_N = 1$ and hence, by induction, all S_k have a representation via a spectral function f_k.

An obvious consequence of the recursion formula is the following recursion:

$$h_k(\lambda) = 2 + \lambda - \frac{1}{h_{k+1}(\lambda)}, \tag{38}$$

with $h_k(\lambda) := \lambda f_k(\lambda) + 1$ and the end condition $h_N(\lambda) = \lambda + 1$.

Now we want to find a filter function g_N to express $u_N = \int g_N(\lambda)\mathrm{d}E_\lambda F^* y$.

Using the expression $I - FK_k = \int (f_{k+1}(\lambda)\lambda + 1)^{-1}\,\mathrm{d}F_\lambda$ we conclude

$$\epsilon_{k+1} = \int (f_{k+1}(\lambda)\lambda + 1)^{-1}\mathrm{d}F_\lambda \epsilon_k = \int \frac{1}{h_{k+1}}\mathrm{d}F_\lambda \epsilon_k = -\int \frac{1}{\Pi_{i=1}^{k+1} h_i(\lambda)}\mathrm{d}F_\lambda y$$

$$v_k = -\int \frac{f_{k+1}(\lambda)}{h_{k+1}(\lambda)}\mathrm{d}E_\lambda F^* \epsilon_k = \int \frac{f_{k+1}(\lambda)}{\Pi_{i=1}^{k+1} h_i(\lambda)}\mathrm{d}E_\lambda F^* y$$

Now we replace $f_{i+1} = 1/\lambda(h_{i+1} - 1)$ and use (29) to obtain

$$\begin{aligned} u_k &= \sum_{i=0}^{k-1} \int_\sigma \frac{1}{\lambda}\frac{h_{i+1}(\lambda) - 1}{\Pi_{j=1}^{i+1} h_j(\lambda)}\mathrm{d}E_\lambda F^* y \\ &= \sum_{i=0}^{k-1} \int_\sigma \frac{1}{\lambda}\left(\frac{1}{\Pi_{j=1}^{i} h_j(\lambda)} - \frac{1}{\Pi_{j=1}^{i+1} h_j(\lambda)}\right)\mathrm{d}E_\lambda F^* y. \\ &= \int \frac{1}{\lambda}\left(1 - \frac{1}{\Pi_{j=1}^{k} h_j(\lambda)}\right)\mathrm{d}E_\lambda F^* y, \end{aligned} \tag{39}$$

where h_k satisfies the backwards recursion formula (38) and the end condition $h_N(\lambda) = \lambda + 1$.

In particular, the N-th iterate, which is our approximate solution, can be expressed as $u_N = \int_\sigma g_N(\lambda)\mathrm{d}E_\lambda F^* y$, with the filter function

$$g_N(\lambda) = \frac{1}{\lambda}\left(1 - \frac{1}{\Pi_{j=1}^{N} h_j(\lambda)}\right). \tag{40}$$

The following theorem yields a representation for g_N in terms of Chebyshev polynomials.

THEOREM 3.3 *Let $T_n(x)$ be the Chebyshev polynomial of the first kind of order n. Then*

$$g_N(\lambda) = \frac{1}{\lambda}\left[1 - \left(\sqrt{\frac{\lambda}{4} + 1}\right)\left(T_{2N+1}\left(\sqrt{\frac{\lambda}{4} + 1}\right)\right)^{-1}\right].$$

Proof Define $p_i(\lambda) := \Pi_{k=N-i}^{N} h_k(\lambda)$, $i = 0, \ldots, N-1$. From the end condition for h_N we find $p_0 = \lambda + 1$. Furthermore, follows from (38)

$$p_{i+1}(\lambda) = h_{N-i-1}(\lambda)p_i(\lambda) = (2+\lambda)p_i(\lambda) - \frac{p_i(\lambda)}{h_{N-i}(\lambda)} = (2+\lambda)p_i(\lambda) - p_{i-1}(\lambda), \tag{41}$$

hence p_i satisfies a three-term recursion. From (38) we see that $p_1 = \lambda^2 + 3\lambda + 1$. If we introduce $p_{-1}(\lambda) := 1$, then the initial conditions $p_{-1}(\lambda)$, $p_0(\lambda)$ together with the three-term recursion (41) completely determine p_i.

We prove the identity

$$p_{N-1}(\lambda) = \frac{T_{2N+1}(\sqrt{(\lambda/4)+1})}{\sqrt{(\lambda/4)+1}} =: q_N(\lambda), \quad \forall N \geq 0.$$

For $N=0$ we have $p_{-1}(\lambda)=1$ and, since $T_1(x)=x$, it follows $q_1=1$. Since $T_3(x)=4x^3-3x$ we find for $N=1$ that $q_2(\lambda)=\lambda+1=p_1(\lambda)$. Hence, the identity $p_{N-1}(\lambda)=q_N(\lambda)$ holds for $N=0,1$. Since two initial conditions and the three-term recursion uniquely determine the sequence $p_i(\lambda)$, $q_i(\lambda)$ we only have to show that q_i satisfies the same recurrence relation as p_i. Note that the following identity holds for all $N \geq 1$ [13]:

$$T_{2N+3}(x) - T_{2N-1}(x) = 2T_{2N+1}(x)T_2(x) = 2T_{2N+1}(x)(2x^2-1).$$

Put $x = ((\lambda/4)+1)^{1/2}$ and multiply the identity by $((\lambda/4)+1)^{-1/2}$ we get

$$\frac{T_{2N+3}(\sqrt{(\lambda/4)+1})}{\sqrt{(\lambda/4)+1}} - \frac{T_{2N-1}(\sqrt{(\lambda/4)+1})}{\sqrt{(\lambda/4)+1}} = \frac{T_{2N+1}(\sqrt{(\lambda/4)+1})}{\sqrt{(\lambda/4)+1}}(\lambda+2).$$

Thus q_N satisfies $q_{N+1}(\lambda)=(\lambda+2)q_N - q_{N-1}$, which is the same recurrence relation as p_n. Hence $q_N = p_{N-1}$. ∎

COROLLARY 3.4 *$g_N(\lambda)$ has the following representations*:

$$g_N(\lambda) = \frac{1}{\lambda}\left(1 - \frac{\cosh\left(\operatorname{arcosh}(\sqrt{(\lambda/4)+1})\right)}{\cosh\left((2n+1)\operatorname{arcosh}\sqrt{(\lambda/4)+1}\right)}\right), \quad \lambda \geq 0 \tag{42}$$

$$g_N(\lambda) = \frac{1}{\lambda}\left(1 - \frac{1}{\sum_{m=0}^{n}\binom{2n+1}{2m}((\lambda/4)+1)^{(n-m)}(\lambda/4)^m}\right). \tag{43}$$

Proof Equation (43) follows from the representation formula for T_{2n+1} [13]:

$$T_{2n+1}(x) = \sum_{m=0}^{n}\binom{2n+1}{2m}x^{2n+1-m}(x^2-1)^m.$$

For the identity (42) we start with the well-known representation [13]

$$T_n(x) = \cos(n \arccos(x)), \quad |x| \leq 1$$

From $\cos(z)=\cosh(iz)$ and $\operatorname{arcosh}(z)=i \arccos(z)$ we get by analytic extension the identity

$$T_n(x) = \cosh(n \operatorname{arcosh}(x)), \quad x \geq 1.$$

From this representation (42) follows, since $\lambda \geq 0$. ∎

The next result concerns the regularization properties of the proposed iterative method.

THEOREM 3.5 *The mapping $y \to u_N$ is a regularization operator, as $N \to \infty$.*

Proof We have to proof the similar properties for the filter function $g_N(\lambda)$ as for the continuous case.

First of all, using L'Hôpital's rule we find

$$\lim_{\lambda\to 0} g_N(\lambda) = -\lim_{\lambda\to 0}\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\sqrt{(\lambda/4)+1}}{T_{2N+1}\sqrt{(\lambda/4)+1}}\right) = -\lim_{z\to 1}\frac{\mathrm{d}}{\mathrm{d}z}\left(\frac{z}{T_{2N+1}(z)}\right)\frac{1}{8\sqrt{(\lambda/4)+1}}\bigg|_{\lambda=0}$$

$$= -\frac{1}{8}\frac{T_{2N+1}(1) - T'_{2N+1}(1)}{T_{2N+1}(1)^2} = \frac{(2N+1)^2-1}{8},$$

where we used $T_n(1)=1$, $T'_n(1) = n^2$. Hence $g_N(\lambda)$ can be extended continuously to $\lambda = 0$.

The estimate $|\lambda_{gN}(\lambda)| \le C$ reduces to

$$\left|1 - \frac{\cosh\left(\operatorname{arcosh}(\sqrt{(\lambda/4)+1})\right)}{\cosh\left((2n+1)\operatorname{arcosh}(\sqrt{(\lambda/4)+1})\right)}\right| \le C,$$

but, by the monotonicity of cosh, it holds that $0 \le (\cosh(x))/(\cosh((2n+1)x)) \le 1$, as a consequence the constant C can be chosen $C=1$.

Finally, $\lim_{N\to\infty} g_N(\lambda) \to (1/\lambda)$ holds, since $\lim_{N\to\infty}\cosh((2N+1)x) = \infty$. ■

We now can prove the convergence rate result similar to the continuous case:

THEOREM 3.6 *Let u_N be defined as above. If the data are exact, $y = Fu^\dagger$ and $u^\dagger$ satisfies a source condition* (27) *for some $\mu > 0$, then*

$$\|u_N - u^\dagger\| \le C_\mu N^{-2\mu}. \tag{44}$$

If the data are contaminated with noise, $\|y - y_\delta\| \le \delta$ and $y = Fu^\dagger$ with $u^\dagger$ satisfying (27), *then we have constants C_μ, C, independent of N, δ, such that*:

$$\|u_{N,\delta} - u^\dagger\| \le C_\mu N^{-2\mu} + C\delta N.$$

The choice $N \sim \delta^{(-1/(2\mu+1))}$ yields the optimal order convergence rates

$$\|u_{N,\delta} - u^\dagger\| \sim \delta^{(2/(2\mu+1))}. \tag{45}$$

Proof We have to find an estimate for

$$|\lambda^\mu(1 - \lambda g_N(\lambda))| \le w_\mu(N), \quad \forall \lambda \ge 0.$$

Hence we need a bound for

$$\xi(\lambda) := \frac{\lambda^\mu \cosh\left(\operatorname{arcosh}\left(\sqrt{(\lambda/4)+1}\right)\right)}{\cosh\left((2N+1)\operatorname{arcosh}\sqrt{(\lambda/4)+1}\right)}, \quad \lambda \geq 0.$$

We may transform the variables $x := (\frac{\lambda}{4}+1)^{1/2}$, $y = \operatorname{arcosh}(x)$ and, using $\cosh(y)^2 - 1 = \sinh(x)^2$, we get

$$\xi(\lambda(x(y))) = \frac{4^\mu \sinh(y)^{2\mu}\cosh(y)}{\cosh((2N+1)y)} =: \zeta(y), \quad y \geq 0.$$

For $y \geq 0$ we may use the addition theorems for cosh:

$$\begin{aligned} |\cosh((2N+1)y)| &= |\cosh(2Ny)\cosh(y) + \sinh(2Ny)\sinh(y)| \\ &= |\cosh(y)\cosh(2Ny)|(1+\tanh(2Ny)\tanh(y))| \geq |\cosh(y)\cosh(2Ny)|, \end{aligned}$$

and, with the estimate $\cosh(x) \geq (1/2)(\exp(x)+1)$, we get

$$|\zeta(y)| \leq 4^\mu \frac{\sinh(y)^{2\mu}}{\cosh(2Ny)} \leq 4^\mu 2 \frac{\sinh(y)^{2\mu}}{\exp(2Ny)+1} =: 4^\mu 2\eta(y)$$

Now differentiation yields the necessary condition for a maximum of η: $(\mu/N)(1+\exp(-x)) = \tanh(x)$. By monotonicity, we see that this equation has a unique solution $x^* > 0$ for $N > \mu$, which must be the maximum of $\eta(y)$, since $\eta(0) = 0$ and $\eta(\infty) = 0$.

Now express $\sinh(x) = \tanh(x)/(\sqrt{1-\tanh(x)^2})$, use $1/(\exp(x)+1) \leq 1$, we get for $N > 2\mu$

$$\eta(x) \leq \left(\frac{\mu}{N}\right)^{2\mu} \frac{(1+\exp(-x^*))^\mu}{\sqrt{1-(\mu^2/N^2)(1+\exp(-x^*))^2}} \leq C\frac{1}{(2N)^{2\mu}}.$$

Hence we get for all μ and $N > 2\mu$

$$\lambda^\mu |1 - \lambda g_N(\lambda)| \leq C\frac{1}{(2N)^{2\mu}},$$

which immediately yields (44) [7].

For a proof of (45) we have to find an estimate

$$g_N(\lambda) \leq C_N, \quad \forall \lambda > 0.$$

Using the same transformation as above, we have to bound for all $y > 0$,

$$\begin{aligned}\phi(y) &:= \frac{\cosh((2N+1)y) - \cosh(y)}{\sinh(y)^2\cosh((2N+1)y)} = \frac{2\sinh((N+1)y)\sinh((N-1)y)}{\sinh(y)^2\cosh((2N+1)y)}\\ &\leq 2\frac{\sinh(Ny)^2}{\sinh(y)^2\cosh(2Ny)} \leq 2\frac{\sinh((N+1)y)^2}{\sinh(y)^2(\cosh(Ny)^2 + \sinh(Ny)^2}\\ &\leq 2\left(\frac{\sinh((N+1)y}{\sinh(y)(\cosh(Ny))}\right)^2 =: 2\psi(y)^2.\end{aligned}$$

Now we may calculate the derivative (using summation formula for sinh, cosh),

$$\psi'(y) = \frac{N}{2}\left(\frac{\sinh(2y) - \frac{1}{N}\sinh(2Ny)}{\sinh(y)^2\cosh(Ny)^2}\right).$$

Now by differentiation it is easy to see that for positive y the function $\sinh(2y) - (1/N)\sinh(2Ny)$ is strictly monotonically decreasing and it vanishes for $y = 0$. Hence ψ has negative derivative for $y > 0$ and $\psi'(0) = 0$. Thus the maximum must be at $y = 0$. By L'Hôpital's rule

$$\psi(0) = \lim_{y\to 0}\frac{\sinh((N+1)y)}{\sinh(y)} = N + 1.$$

Hence $|g_N(\lambda)| \leq 2(N+1)^2 \leq CN^2$, with a constant C independent of N. With the results of [7, Theorem 4.3] the proof is finished. ■

4. Numerical experiments

We are now concerned with the numerical realization of the described algorithm. We consider the discrete variant (19, 20) and a discretization of the continuous algorithm (10, 12).

The first one has a straightforward implementation. For the continuous approach we use an explicit time-discretization $Q'(t) \sim (1/\Delta t)(Q_{n+1} - Q_n)$. Then equation (10) becomes an iterative procedure: (note that the Riccati-equation has to be solved backwards in time)

$$\begin{aligned}&Q_n = Q_{n+1} + \Delta t(I - Q_{n+1}FF^*Q_{n+1}), \quad n = N-1, \ldots, 0\\ &Q_N = 0.\end{aligned}$$

Equation (12) is discretized in a similar manner:

$$u_{n+1} = u_n - \Delta t(F^*Q_n(Fu_n - y)), \quad n = 0, \ldots, N-1$$

together with some initial condition u_0.

A more efficient method is to use a recursion for $B_n := F^* Q_n$. Since Q_n is symmetric, then

$$B_n = B_{n+1} + \Delta t F^*(I - B_{n+1}^* B_{n+1}). \tag{46}$$

Hence, we get

$$u_{n+1} = u_n - \Delta t(B_n(Fu_n - y)). \tag{47}$$

Since we used an explicit discretization scheme, the method will be only stable if we bound the stepsize appropriately, e.g., $\Delta t\|F^*F\| \leq 1$. The explicit discretization has the advantage that no matrix inversion is needed, by paying the price of a restricted stepsize. A detailed analysis of the regularization properties of this iterative scheme, in the spirit of section 3, is of course also possible.

As a benchmark problem we consider an integral equation of the first kind:

$$Fu = \int_0^1 k(x, y)u(y)\mathrm{d}y.$$

For a discretization of this operator, we split the unit interval $I = [0, 1]$ into m subintervals and discretize u by using a uniform discretization with piecewise linear, continuous splines on each subinterval (also known as Courant-finite elements). The integral is evaluated by the trapezoidal rule one each subinterval. As evaluation points for x we used $x_i = i/m$, $i = 0, \ldots, m$. This results in a $(m+1) \times (m+1)$ matrix equation:

$$F_m u_m = y_m. \tag{48}$$

We tested our algorithms with F replaced by the discretized version F_m.

We do not address the question how the discretization parameter m has to be related to the regularization parameter (the iteration index in our case), but we simply consider the discretized equation as the given ill-posed problem. Hence we use the Euclidean norm in R^{m+1} on the discrete variables u_m, y_m.

For our numerical test we used two different kernel functions $k(x, y)$:

$$k_1(x, y) := \begin{cases} \left(1 - \dfrac{(x-y)^2}{0.1}\right)^6 & \text{if } (x-y)^2 \leq 0.1 \\ 0 & \text{else} \end{cases} \tag{49}$$

$$k_2(x, y) := \frac{1}{2\sqrt{20}} \exp\big(-20(x-y)^2\big). \tag{50}$$

The first one is 6-times continuously differentiable and hence leads to a mildly ill-posed problem. The second one $k_2(x, y)$ is smooth, hence it leads to an exponentially ill-posed problem.

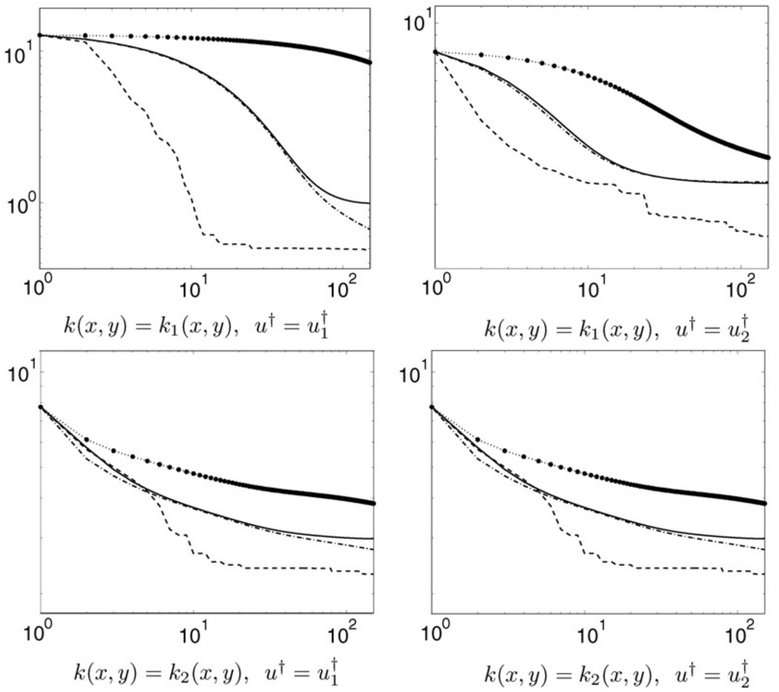

Figure 1. Evolution of the error $\|u_N - u^\dagger\|$ for exact data for all four algorithms.

We tested our methods for two exact solutions

$$u_1^\dagger(x) := x(1-x) + \cos(20x), \quad u_2^\dagger(x) = \begin{cases} 1 & \text{if } 0.3 \le x \le 0.5 \\ 0 & \text{else} \end{cases}$$

We compared both algorithms with the Landweber-iteration and the CG-method [7]. Throughout our numerical experiments we used a discretization of $m = 300$.

Figure 1 shows the error $\|u_N - u^\dagger\|$ over the iteration index N on a log–log scale for the four algorithms and the different choices of $u^\dagger$ and $k(x, y)$. Here the full line corresponds to the discrete dynamic programming method, the dotted line to the Landweber iteration, the dashed dotted to the continuous method with explicit time discretization, and the dashed line to the conjugate-gradient method.

Furthermore we contaminated the data with 10% random noise. The results are shown in figure 2. Since in this case the iteration cannot converge, a correct stopping criterion would be necessary. An *a priori* stopping criterion was derived in Theorems 3.1 and 3.6. Of course *a posteriori* stopping criteria are more flexible. A more detailed analysis of these rules (e.g., Morozov's discrepancy principle, or the Engl–Gfrerer-type rules [7]) are out of the scope of this work.

We observe that the two methods based on dynamic programming techniques are almost similar. Moreover, these two methods have about the same convergence rates as the conjugate-gradient algorithm, indicated by the same slope of the lines. This is confirmed by theory, as the number of iterations k to reach a certain noise level δ under a source condition is $k \sim \delta^{(-1/(2\nu+1))}$ both for CG [7, Theorem 7.13] and the

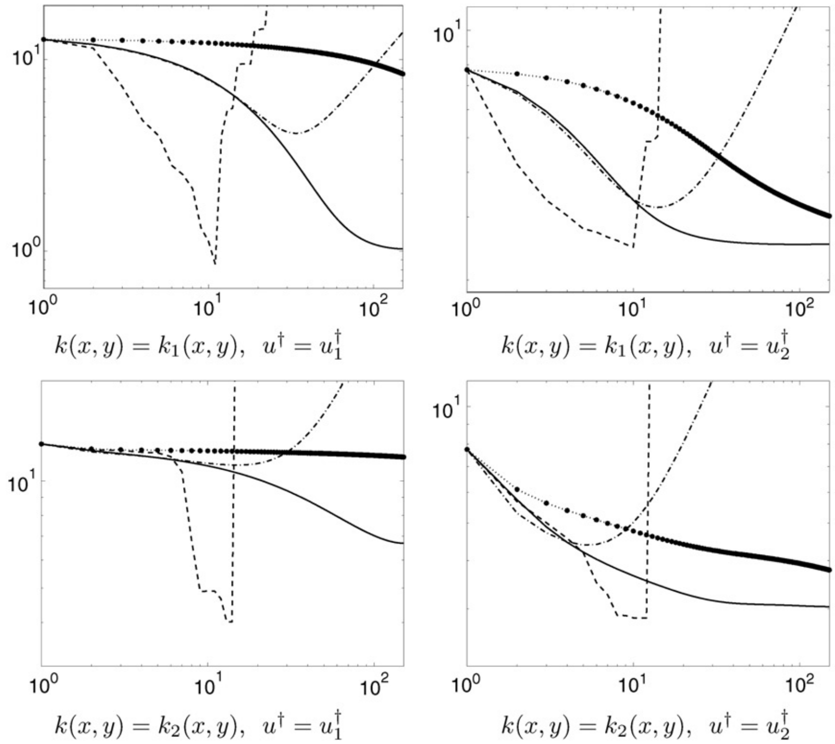

Figure 2. Evolution of the error $\|u_N - u^\dagger\|$ for noisy data for all four algorithms.

dynamic programming techniques (Theorem 3.6), whereas for Landweber iteration it is larger, namely $k \sim \delta^{(-2/(2\nu+1))}$. Note also, that CG is only a regularization method together with a discrepancy principle and is not one in the sense of [7] if the noise level vanishes. Such a phenomenon does not happen for the dynamic programming iterations.

Let us report on the overall costs of computation. Let F be a matrix of size $n \times m$. Then if N time-steps (or iteration steps) are made, the complexity for Landweber iteration and CG are $\mathcal{O}(nmN)$, since only matrix-vector multiplications have to be performed. The bottleneck for the dynamic programming iterations (19)–(20) and (46)–(47) is the Riccati equation. Since in each step a matrix–matrix product has to be computed we end up with an overall complexity for the implicit scheme (19)–(20) of $\mathcal{O}(n^2mN + m^2nN + n^3N)$ and $\mathcal{O}(n^2mN + m^2nN)$ for the explicit one (46)–(47). This shows that these iterations have a complexity of at least one power higher than other iterations. If $n \sim m$, then the explicit and the implicit dynamic iterations are even of comparable complexity. In this case the implicit version is to be favored as it has no stepsize restrictions.

5. Final remarks and conclusions

In this article, we combined control theory with abstract regularization theory. We proposed iterative algorithms for solving linear inverse problems in Hilbert spaces and scrutinized their regularization properties. Our algorithms give rise to convergence and convergence rates under the standard source conditions. The convergence properties are comparable to a conjugate gradient method.

However, we have to admit, that in terms of computational complexity our method is not really competitive with standard methods, as it involves matrix-matrix products in each iteration. On the other hand, the most costly part of our computation, the computation of $Q(t)$ can be performed independent of the data. Hence, if for a fixed operator the same problem has to be solved with different data, then $Q(t)$ only has to be computed once, e.g., by (46) and the remaining iteration (47) involving the data is of similar complexity as the usual iteration methods. In this case our iterations are competitive with CG.

Most of all, we consider this work a good starting point into further directions: first of all it should be noticed that, if Q_n is chosen constant, and not computed by the Riccati equation, the continuous regularization method proposed in this article reduces to a preconditioned Landweber iteration. Therefore, the dynamic programming regularization method can be considered as a generalization of the Landweber method. Since the Landweber method is convergent we expect that solving the Riccati equation is a numerical overkill. Instead one can think of solving the equation inexact or using just a few number of steps of the Riccati iteration to get a matrix Q, which can be used in a preconditioned Landweber (or CG) iteration.

Secondly, we expect that the real power of the combination of control theory and regularization comes into play when considering dynamical inverse problems, that is, if the data or the operator depend on time. In this case standard iterations cannot be used, but the dynamic programming principle still can be applied.

Acknowledgments

The work of S.K. is supported by Austrian Science Foundation under grant SFB F013/F1317; the work of A.L. is supported by the Austrian Academy of Sciences and by CNPq, grant 306020/06-8 and 478099/04-5.

References

[1] Adams, R.A., 1975, *Sobolev Spaces* (New York: Academic Press).

[2] Bellman, R., 1953, *An Introduction to the Theory of Dynamic Programming* (California, Santa Monica: The Rand Corporation).

[3] Bellman. R., 1957, *Dynamic Programming* (Princeton, N.J: Princeton University Press).

[4] Bellman, R., Dreyfus, S.E. and Stuart, E., 1962, *Applied Dynamic Programming* (Princeton, NJ: Princeton University Press).

[5] Bellman, R. and Kalaba, R., 1965, *Dynamic Programming and Modern Control Theory* (New York/London: Academic Press).

[6] Dreyfus, S.E., 1965, *Dynamic Programming and the Calculus of Variations* (New York/London Academic Press).

[7] Engl, H.W., Hanke, M. and Neubauer, A., 1996, *Regularization of Inverse Problems* (Dordrecht: Kluwer Academic Publishers).

[8] Engl, H.W., Kunisch, K. and Neubauer, A., 1989, Convergence rates for Tikhonov regularization of nonlinear ill-posed problems. *Inverse Problems*, **5**, 523–540.

[9] Engl, H.W. and Scherzer, O., 2000, *Convergence Rates Results for Iterative Methods for Solving Nonlinear Ill-posed Problems*, Surveys on Solution Methods for Inverse Problems (Vienna: Springer), pp. 7–34.

[10] Hanke, M., Neubauer, A. and Scherzer, O., 1995, A convergence analysis of the Landweber iteration for nonlinear ill-posed problems. *Numerische Mathematik*, **72**, 21–37.

[11] Kurzhanskiĭ, A.B. and Sivergina, I.F., 1998, The dynamic programming method in inverse estimation problems for distributed systems. *Doklady Mathematics*, **53**, 161–166.

[12] Kurzhanskiĭ, A.B. and Sivergina, I.F., 1999, Dynamic programming in problems of the identification of systems with distributed parameters. *Journal of Applied Mathematics and Mechanics*, **62**, 831–842.

[13] Magnus, W., Oberhettinger, F. and Soni, R.P., 1966, *Formulas and Theorems for the Special Functions of Mathematical Physics* (Heidelberg, Berlin: Springer).

[14] Morozov, V.A., 1993, *Regularization Methods for Ill–Posed Problems* (Boca Raton: CRC Press).

[15] Tautenhahn, U., 1994, On the asymptotical regularization of nonlinear ill-posed problems. *Inverse Problems*, **10**, 1405–1418.