Solution of Ill-posed Problems with Chebfun

A. Alqahtani · T. Mach · L. Reichel

May 23, 2022

Abstract The analysis of linear ill-posed problems often is carried out in function spaces using tools from functional analysis. However, the numerical solution of these problems typically is computed by first discretizing the problem and then applying tools from finite-dimensional linear algebra. The present paper explores the feasibility of applying the Chebfun package to solve ill-posed problems with a regularize-first approach numerically. This allows a user to work with functions instead of vectors and with integral operators instead of matrices. The solution process therefore is much closer to the analysis of ill-posed problems than standard linear algebra-based solution methods. Furthermore, the difficult process of explicitly choosing a suitable discretization is not required.

Department of Mathematics, King Khalid University, P.O. Box 9004, Abha, Saudi Arabia, and Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: `aalqah11@gmail.com`.

Institute for Mathematics, University of Potsdam, 14476 Potsdam, Germany. E-mail: `mach@uni-potsdam.de`. Significant parts of the research have been carried out while the second author was with the Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: `reichel@math.kent.edu`.

1 Introduction

We are interested in the solution of Fredholm integral equations of the first kind,

$$\int_{\Omega_1} \kappa(s,t)x(t)\,dt = g(s), \qquad s \in \Omega_2, \tag{1.1}$$

with a square integrable kernel κ. The Ω_i are subsets of $\mathbb{R}^{d_i}$ for $i = 1, 2$, where d_i is a small positive integer. Such integral equations are common in numerous applications including remote sensing, computerized tomography, and image restoration.

Two major problems arise when solving (1.1): i) The function space is of infinite dimensionality, and ii) small changes in g may correspond to large changes in x as exemplified by

$$\max_{s \in \Omega_2} |\int_{\Omega_1} \kappa(s,t)\cos(2\pi mt)\,dt|, \qquad \Omega_1 = \Omega_2 = [0, 1],$$

where the maximum can be made tiny by choosing m large, despite the fact that the maximum of $|\cos(2\pi mt)|$ is one. This is a consequence of the Riemann–Lebesgue theorem; see, e.g., [7,12] or below for discussions of this result. The second problem is particularly relevant when the available right-hand side g is a measured quantity subject to observational errors, as is the case in many applications.

Usually one deals with problem i) by first discretizing the functions $x(t)$ and $g(s)$ in (1.1) using, for instance, n piece-wise constant, linear, or polynomial basis functions; see e.g., [11] or [13]. The kernel $\kappa(s,t)$ is discretized analogously. This transforms the integral equation (1.1) into a linear system of algebraic equations. Problem ii) causes the coefficient matrix of said system to be ill-conditioned when n is large. Straightforward solution of this linear system of equations generally is not meaningful because of severe error propagation. Therefore, the linear system has to be regularized. This can, for instance, be achieved by Tikhonov regularization or truncated singular value decomposition (TSVD). While the first dampens the influence of small singular values, the latter outright ignores them. One is then often faced with a trade-off between a small discretization error and a small error caused by the regularization; see, e.g., Natterer [18]. In fact, often the more basis functions are used for the discretization, the more ill-conditioned the resulting coefficient matrix becomes, and the larger the need of regularization.

In this paper, we first regularize and then discretize the problem. Regularization is achieved by modifying the singular value expansion (SVE) of the kernel. This expansion provides an excellent starting point for discretizing the problem. The discretized problem is a linear system of equations with a diagonal matrix. This system can be solved trivially. Furthermore, a user does not have to choose the number of discretization points. Regularizing first simplifies the discretization.

We will compute the SVE of the kernel using Chebfun [6], which is a software package that simulates working with functions in MATLAB. Chebfun

approximates functions by piece-wise polynomials. The computed solution is a piece-wise polynomial approximation of the desired solution $x(t)$ of (1.1). The advantage of using Chebfun is that the computed solution will feel and behave like a function. Therefore, our approach is arguably closer to directly solving (1.1) than to discretizing the integral equation before solution.

This paper is organized as follows. In the second section, we provide basic definitions, introduce our notation, and briefly discuss Chebfun and the singular value expansion. Section 3 discusses the truncated singular value expansion method (TSVE) for the solution of ill-posd problems, and the Tikhonov regularization method is described in Section 4. Numerical results that illustrate the performances of these methods are reported in Section 5. After we have established that this regularize-first approach works for linear ill-posed problems in one space-dimension, we will extend the ideas to problems in two space-dimensions in Section 6. Concluding remarks can be found in Section 7.

We note that other approaches to work with functions are described in the literature; see, e.g., Yarvin and Rokhlin [25]. We are not aware of discussions of this approach to the solution of linear ill-posed problems. The availability of Chebfun [6] makes the regularize-first technique easy to implement.

2 Basics

Let $L^2(\Omega_i)$ for $i = 1, 2$ be spaces of Lebesgue measurable square integrable functions with inner products

$$\langle a, b \rangle_{\Omega_i} = \int_{\Omega_i} \overline{a(t)} \, b(t) \, dt, \quad \text{for } i = 1, 2, \tag{2.1}$$

where $\overline{a(t)}$ represents the complex conjugate of $a(t) \in \mathbb{C}$. Based on these inner products, we define the L^2-norms

$$\|f\|_{\Omega_i} = \langle f, f \rangle_{\Omega_i}^{1/2}, \quad \text{for } i = 1, 2.$$

The spaces $L^2(\Omega_i)$ for $i \in \{1, 2\}$ equipped with the norms above are complete vector spaces, i.e., they are *Hilbert space*; see, e.g., [11].

A given kernel $\kappa(\cdot, \cdot) \in L^2(\Omega_1 \times \Omega_2)$ induces the bounded linear operator $A : L^2(\Omega_1) \to L^2(\Omega_2)$ defined by

$$(Ax)(s) = \int_{\Omega_1} \kappa(s, t) x(t) \, dt; \tag{2.2}$$

see, e.g., [11, Thm. 3.2.7]. The operator is sometimes called a *Hilbert-Schmidt integral operator* and the kernel κ a *Hilbert-Schmidt kernel*. We can write (1.1) as

$$Ax = g. \tag{2.3}$$

We assume here that g is in the range of A. If A has a nontrivial null space, then we are interested in the solution of (2.3) of minimal norm. We refer to this solution as x_{exact}.

In applications that arise in the sciences and engineering, the right-hand side g of (1.1) often is a measured quantity and therefore is subject to observational errors. We are therefore interested in the situation when the error-free function g is not available, and only an error-contaminated approximation $g^\delta \in L^2(\Omega_2)$ of g is known. We assume that g^δ satisfies

$$\left\| g - g^\delta \right\|_{\Omega_2} \leq \delta,$$

with a *known* bound $\delta > 0$. The solution of the equation

$$Ax = g^\delta, \quad \text{with } x \in L^2(\Omega_1) \text{ and } g^\delta \in L^2(\Omega_2), \qquad (2.4)$$

generally is not a meaningful approximation of the desired solution x_{exact} of (2.3), since A is not continuously invertible. In fact, equation (2.4) might not even have a solution.

The operator A depends on the kernel κ. We will now take a closer look at known theory about the kernel function κ. For any square integrable kernel κ, we define the singular value expansion (SVE) [21, §4] as

$$\kappa(s,t) = \sum_{i=1}^{\infty} \sigma_i \phi_i(s) \psi_i(t) \qquad (2.5)$$

with convergence in the L^2-sense. The functions $\phi_i \in L^2(\Omega_2)$ and $\psi_i \in L^2(\Omega_1)$ are referred to as singular functions. These functions are orthonormal with respect to the appropriate inner product (2.1) [21, §5], i.e.,

$$\langle \psi_i, \psi_j \rangle_{\Omega_1} = \langle \phi_i, \phi_j \rangle_{\Omega_2} = \delta_{ij}, \quad \text{for } i,j = 1,2,\dots .$$

The quantities σ_i are known as singular values. They are nonnegative and we assume them to be ordered non-increasingly,

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq 0.$$

It can be shown that the only limit point of the singular values for square integrable kernels is zero [21, §5].[1]

Let the series (2.5) be uniformly convergent for $s \in \Omega_2$ and $t \in \Omega_1$. Then, as shown in [21, §8], the series converges point-wise. When the summation is finite, the kernel κ is said to be *separable* (or *degenerate*). Most applications do not have a separable kernel. However, square integrable kernels can be approximated well by such a kernel, which is obtained by truncating the number of terms in (2.5) to a suitable finite number ℓ. Let

$$\kappa_\ell(s,t) = \sum_{i=1}^{\ell} \sigma_i \phi_i(s) \psi_i(t). \qquad (2.6)$$

This is the closest kernel of rank at most ℓ to κ in the L^2-norm [21, §18 Approximation Theorem]. We will use this result to justify the application of the truncated singular value expansion method (TSVE) discussed in Section 3.

---

[1]Schmidt calls the singular values eigenvalues, since he is mainly concerned with symmetric kernels and the concept of singular values was not developed when he published his paper. We follow modern notation here.

We also will be using the Approximation Theorem to restrict our expansion to singular values that are larger than ε, where ε is a small enough cut-off, say 10^{-8} or 10^{-16}. There is a trade-off between computing time and approximation accuracy. We try to choose ε far below the regularization error so that our choice does not have a significant effect on the accuracy. At the same time, a small ε means higher cost for computing the singular value expansion and forming the computed approximate solution.

In this paper, we will use two regularization methods, TSVE and Tikhonov regularization, to compute an approximate solution of (2.3). The right-hand side g is assumed not to be available; only a perturbed version g^δ is assumed to be known; cf. (2.4).

The TSVE method is based on the Approximation Theorem mentioned above. Thus, we approximate the kernel κ by κ_ℓ for some suitable $\ell \geq 0$. This results in an approximation A_ℓ to A. We are interested in computing the solution x_ℓ of minimal norm of the problem

$$\min_{x \in L^2(\Omega_1)} \left\| A_\ell x - g^\delta \right\|_{\Omega_2}, \tag{2.7}$$

where

$$(A_\ell x)(s) = \int_{\Omega_1} \kappa_\ell(s,t) x(t)\, dt, \quad s \in \Omega_2.$$

The parameter ℓ is a regularization parameter that determines how many singular values and basis functions of κ are used to compute the approximate solution x_ℓ of (2.4). The remaining singular values, which are smaller than or equal to σ_ℓ, are ignored.

Tikhonov regularization replaces the system (2.4) by the penalized least-squares problem

$$\min_{x \in L^2(\Omega_1)} \left\{ \left\| A x - g^\delta \right\|_{\Omega_2}^2 + \lambda^2 \|x\|_{\Omega_1}^2 \right\}, \tag{2.8}$$

which has a unique solution, denoted by x_λ, for any nonvanishing value of the regularization parameter λ. Substituting the SVE (2.5) into (2.8) shows that Tikhonov regularization dampens the contributions to x_λ of singular functions with large index i the most; increasing $\lambda > 0$ results in more damping. Since we cannot deal with an infinite series expansion, we will, in practice, first cut-off all singular values that are less than ε as explained above, and then apply Tikhonov regularization. This is sometimes referred to as a discretization by kernel approximation [11, Sect. 4.2].

The determination of suitable values of the regularization parameters, ℓ in (2.7) and λ in (2.8), is important for the quality of the computed approximate solution. Several methods have been described in the literature including the discrepancy principle, the L-curve criterion, and generalized cross validation; see [4, 8, 15, 16, 19, 20] for recent discussions on their properties and illustrations of their performance. Regularization methods typically require that regularized solutions for several values of the regularization parameter be computed and compared in order to determine a suitable approximate solution of (2.4).

2.1 Chebfun

We solve (1.1) by first regularizing followed by discretization. However, we still want to compute the solution numerically. Thus, we need a numerical library that can handle functions in an efficient way. Since a function $t \mapsto f(t)$ represents uncountable many pairs of $\{t, f(t)\}$, a computer only can handle approximations to functions numerically.[2]

We chose the MATLAB package Chebfun [6] for this purpose. Chebfun uses piece-wise polynomials anchored at Chebyshev points, so-called *chebfuns*, to approximate functions. All computations within Chebfun's framework are done with these approximations of actual functions. This means that we project the functions in $L^2(\Omega_2)$ onto a space of piece-wise Chebyshev polynomials over Ω_2. Thus, Chebfun simulates computation with functions that are approximated. One may argue that this is a discretization. However, Chebfun's framework is significantly different from other discretizations in the sense that it gives a user the feeling of computing with functions. In particular, a user does not explicitly have to determine a discretization or where to split functions into polynomial pieces.

Chebfun's functionality includes the computation of sums and products of functions and derivatives, inner products, norms, and integrals. Chebfun2/3, Chebfun's extensions to functions of two and three variables, also can compute outer products and, most importantly for us here, the singular value expansion [23]. The algorithm behind the singular value expansion uses a continuous analogue of adaptive cross approximation: The approximation of $\kappa(s, t)$ is computed by an iterative process. First, an approximation of a maximum point $(\hat{s}, \hat{t})$ of $\kappa(s, t)$ is determined. The exact computation of a maximum point is not important, and the maximum point is not required to be unique. The function is then approximated by

$$\kappa_1(s, t) = \frac{\kappa(s, \hat{t})\kappa(\hat{s}, t)}{\kappa(\hat{s}, \hat{t})},$$

where $s \mapsto \kappa(s, \hat{t})$ and $t \mapsto \kappa(\hat{s}, t)$ are one-dimensional chebfuns in s and t, respectively. This process is then repeated for $\kappa(s, t) - \kappa_1(s, t)$ to find a rank-1 approximation of the remainder. By recursion one obtains after k steps a rank-k approximation of the original kernel. As soon as the remainder is sufficiently small, the computed rank-k approximation is the sought approximation to $\kappa(s, t)$. At the end we have $\kappa(s, t) \approx C(s)MR(t)^T$, with $C(s)$ and $R(t)$ row vectors of functions, and M a dense matrix of size $k \times k$.

Based on this approximation, it is easy to compute the singular value expansion. Chebfun's continuous analogue of the QR factorization can be used to find orthogonal bases for $C(s)$ and $R(t)$,

$$C(s) = Q_c(s)R_c, \qquad R(t) = Q_r(t)R_r,$$

---

[2]There are some notable exceptions such as $f(t) = t$. However, we cannot assume that the solution of (1.1) will fall into this very small set of functions.

where the columns of Q_c and Q_r are orthonormal functions and the $k \times k$ matrices R_c and R_r are upper triangular. Let

$$\widetilde{M} = R_c M R_r^T$$

and compute the singular value decomposition $\widetilde{M} = U \Sigma V^T$. Then

$$\kappa(s,t) \approx Q_c(s) U \Sigma V^T Q_r(t)^T$$

is the desired singular value expansion of the approximation of the kernel. The singular functions are the columns of $Q_c(s)U$ and $Q_r(t)V$; see [23] for further details. A very similar process, known as adaptive cross approximation [2,3], was used in [17] for the discrete case of matrices and vectors.

Chebfun has some limitations. Currently only functions of at most three variables can be approximated by Chebfun. Hence, we are limited to ill-posed problems in one space-dimension and to problems in higher dimension with a special structure. We will in Section 6 discuss problems in two space-dimensions with special kernels that can be handled by the present version of Chebfun. In higher dimensions, Chebfun is limited to domains that are tensor products of intervals, disks, spheres, or solid spheres. In this paper, all domains are rectangles. Chebfun also needs multivariate functions to be of low rank for an efficient approximation, i.e., there has to exist a sufficiently accurate separable approximation. This is for instance not the case for the kernel $\kappa(s,t) = st - \min(s,t)$ from the `deriv2` example of the Regularization Tools package [13]. This limits the application of the methods described in this paper. However, the Chebfun package is still under development and some of the limitations mentioned might not apply to future releases.

3 The TSVE method

Assume that the kernel can be expressed as

$$\kappa(s,t) = \sum_{i=1}^{\infty} \sigma_i \phi_i(s) \psi_i(t) \tag{3.1}$$

and is not separable, and that the solution of (1.1) can be written as

$$x(t) = \sum_{j=1}^{\infty} \beta_j \psi_j(t). \tag{3.2}$$

The fact that κ is non-separable implies that all σ_i are positive, and the assumption that the solution is of the form (3.2) essentially states that the solution has no component in the null space of A. This assumption is justified since the null space of A is orthogonal to all the ψ_j and, thus, a component in the null space would increase the norm of the solution, but not help with the approximation of (1.1).

Substituting (3.1) and (3.2) into (1.1), and using the orthonormality of the basis functions yields

$$\sum_{i=1}^{\infty} \sigma_i \beta_i \phi_i(s) = g(s).$$

We further probe the equation with $\phi_k(s)$ for all k and use the orthonormality of the basis functions to obtain

$$\sigma_k \beta_k = \int_{\Omega_2} \phi_k(s) g(s) \, ds, \qquad \text{for all } k.$$

Thus, the exact solution of (2.3) is given by

$$x(t) = \sum_{j=1}^{\infty} \beta_j \psi_j(t), \quad \text{with } \beta_j = \frac{\int_{\Omega_2} \phi_j(s) g(s) \, ds}{\sigma_j}. \tag{3.3}$$

Truncating this series after ℓ terms and using the error-contaminated right-hand side g^δ instead of g, we obtain the TSVE solution of (2.4),

$$x_\ell(t) = \sum_{j=1}^{\ell} \beta_j^\delta \psi_j(t), \quad \text{with } \beta_j^\delta = \frac{\int_{\Omega_2} \phi_j(s) g^\delta(s) \, ds}{\sigma_j}. \tag{3.4}$$

The truncation parameter ℓ can be chosen as needed. Its choice will be discussed below.

The following lemma links the projection of the error onto the space spanned by the ϕ_i to the bound δ for the norm of the error in the data.

Lemma 3.1 *Let $n(s) = g(s) - g^\delta(s)$ with $\|n\|_{\Omega_2} \le \delta$. Then,*

$$\sum_{i=1}^{\infty} \left(\int_{\Omega_2} \phi_i(s) n(s) \, ds \right)^2 \le \delta^2, \tag{3.5}$$

where the ϕ_i are orthonormal basis functions determined by the SVE of the kernel κ.

Proof Using the basis functions ϕ_i, the error n can be represented as

$$n(s) = \sum_{j=1}^{\infty} \gamma_j \phi_j(s) + \phi^\perp(s)$$

for certain coefficients γ_j, where the function $\phi^\perp$ is orthogonal to all the functions ϕ_j. It follows that

$$\int_{\Omega_2} \phi_i(s) n(s) \, ds = \int_{\Omega_2} \phi_i(s) \left(\sum_{j=1}^{\infty} \gamma_j \phi_j(s) + \phi^\perp(s) \right) \, ds.$$

The orthogonality of the basis functions ϕ_j allows us to simplify the above expression to

$$\int_{\Omega_2} \phi_i(s) n(s)\, ds = \gamma_i.$$

The same argument can be used to show that

$$\sum_{j=1}^{\infty} \gamma_j^2 \leq \|n\|_{\Omega_2}^2 \leq \delta^2.$$

Combining these results shows (3.5). $\qquad\square$

4 Tikhonov regularization

Instead of solving (2.3) exactly, we solve the minimization problem

$$\min_{x \in L^2(\Omega_1)} \left\{ \left\| Ax - g^\delta \right\|_{\Omega_2}^2 + \lambda^2 \|x\|_{\Omega_1}^2 \right\}, \tag{4.1}$$

where the regularization parameter $\lambda > 0$ typically is determined during the solution process. This parameter balances the influence of the norm of the residual $Ax - g^\delta$ and the norm of the solution x. Using the definition of the L^2-norm, equation (4.1) can be written as

$$\min_{x \in L^2(\Omega_1)} \left\{ \int_{\Omega_2} \left| Ax - g^\delta \right|^2 ds + \lambda^2 \int_{\Omega_1} |x|^2 dt \right\}. \tag{4.2}$$

This minimization problem can be solved in a variety of ways. For instance, one could apply a continuous version of Golub–Kahan bidiagonalization to the operator A. Then A is reduced to (an infinite) bidiagonal matrix. The bidiagonalization process can be truncated as soon as a solution of the reduced problem that satisfies the discrepancy principle has been found; this approach is analogous to the computations described in [5] for discretized problems. A straightforward way to solve (4.2), though not necessarily the fastest, is to determine the minimizer by applying the SVE (2.5). We will use this approach in the computed examples. Thus, substituting (3.1) and (3.2) into (4.2), and using the orthonormality of the basis functions, we obtain

$$\min_{x \in L^2(\Omega_1)} \sum_{i=1}^{\infty} \left(\sigma_i^2 \beta_i^2 - 2\sigma_i \beta_i \int_{\Omega_2} \phi_i(s) g^\delta(s)\, ds + \lambda^2 \beta_i^2 \right) + \int_{\Omega_2} \left| g^\delta(s) \right|^2 ds.$$

We now can compute the solution as

$$x_\lambda(t) = \sum_{j=1}^{\infty} \beta_j^{(\lambda)} \psi_j(t), \quad \text{with } \beta_j^{(\lambda)} = \frac{\sigma_j \int_{\Omega_2} \phi_j(s) g^\delta(s)\, ds}{(\sigma_j^2 + \lambda^2)}. \tag{4.3}$$

Table 5.1: 1D examples used for the numerical experiments.

Example	Domain	$\kappa(s,t)$, $x(t)$ and $g(s)$	Reference
Baart	$\Omega_1 = [0,\pi]$ $\Omega_2 = [0,\pi/2]$	$\kappa(s,t) = \exp(s\cos(t))$ $x(t) = \sin(t)$ $g(s) = 2\sinh(s)/s$	[1, Ex. 4.2]
Fox-Goodwin[3]	$\Omega_1 = [0,1]$ $\Omega_2 = [0,1]$	$\kappa(s,t) = (s^2 + t^2)^{\frac{1}{2}}$ $x(t) = t$ $g(s) = \frac{1}{3}((1+s^2)^{\frac{3}{2}} - s^3)$	[9, p. 520]
Gravity	$\Omega_1 = [0,1]$ $\Omega_2 = [0,1]$	$\kappa(s,t) = (1 + (s-t)^2)^{-\frac{3}{2}}$ $x(t) = \sin(\pi t) + \frac{1}{2}\sin(2\pi t)$	[24, p. 17]
Shaw	$\Omega_1 = [-\frac{\pi}{2}, \frac{\pi}{2}]$ $\Omega_2 = [-\frac{\pi}{2}, \frac{\pi}{2}]$	$\kappa(s,t) = (\cos(s) + \cos(t))\left(\frac{\sin(u)}{u}\right)^2$ $u = \pi(\sin(s) + \sin(t))$ $x(t) = 2e^{-6(t-0.8)^2} + e^{-2(t+0.5)^2}$	[22, p. 97]
Wing	$\Omega_1 = [0,1]$ $\Omega_2 = [0,1]$	$\kappa(s,t) = t\exp(-s\,t^2)$ $x(t) = \mathbf{1}_{[\frac{1}{3}, \frac{2}{3}]}$ $g(s) = \frac{\exp(-s/9) - \exp(-4s/9)}{2s}$	[24, p. 109]

5 Numerical experiments

In this section, we illustrate the performance of the methods described in Sections 3 and 4 by reporting some numerical results. In particular, we compare the performance of the regularize-first approaches based on Chebfun discussed in this paper to more commonly used discretize-first approach. Both approaches have their advantages and the regularize-first approach can be expected to be slower due to the effort required to work with chebfuns. The examples of this section show this to be the case, but usually not by very much. This indicates that the regularize-first approach based on Chebfun is a practical solution method for many linear ill-posed problems.

We first consider five test problems in one space-dimension. These problems are from Regularization Tools by Hansen [13]. In the next section, we will follow up with two 2D problems, an image deblurring problem with Gaussian blur and a diffusion problem inspired by IR Tools [10]. All computations were carried out in MATLAB R2020b running on a computer with two CPUs: Intel(R) Xeon(R) E5-2683 v4@2.10 GHz processor with 512 GB of RAM. There are two MATLAB Live Scripts accompanying this paper that showcase the most important of the following numerical experiments.

Each test problem from Regularization Tools by Hansen [13] provides us with an integral equation of the form (1.1). These test problems are listed in Table 5.1. For two examples, Gravity and Shaw, the function $g(s)$ is not explicitly given in the table, and is instead computed by evaluating the integral (1.1) with Chebfun. In example Wing, the expression $\mathbf{1}_{[\frac{1}{3}, \frac{2}{3}]}$ denotes the

indicator function, which is 1 for all points in $[\frac{1}{3}, \frac{2}{3}]$ and 0 otherwise. We compare the regularize-first approach using Chebfun-based computations with the discretize-first approach, where the discretizations are determined by MAT-LAB functions in Regularization Tools [13].

For the discretize-first approach, these problems are discretized by a Nyström method or a Galerkin method with orthogonal test and trial functions to give a linear system of equations $\tilde{A}\boldsymbol{x} = \boldsymbol{g}$, where $\tilde{A} \in \mathbb{R}^{n \times n}$ is the discretized integral operator, $\boldsymbol{x} \in \mathbb{R}^n$ is a discretization of the exact solution x_{exact}, and $\boldsymbol{g} \in \mathbb{R}^n$ is the corresponding error-free right-hand side vector. We generate the error-contaminated vector $\boldsymbol{g}^\delta \in \mathbb{R}^n$ according to

$$\boldsymbol{g}^\delta = \boldsymbol{g} + \alpha \frac{\|\boldsymbol{g}\|_2}{\|\boldsymbol{e}\|_2} \boldsymbol{e},$$

where $\boldsymbol{e} \in \mathbb{R}^n$ is a random vector whose entries are from a normal distribution with mean zero and variance one. The parameter $\alpha > 0$ is the noise level.

For the regularize-first approach, we use the MATLAB package Chebfun [6] to represent the kernel $\kappa(s, t)$, the function $g(s)$ that represents the error-free right-hand side, and the desired solution $x(t)$. We define the error-contaminated function $g^\delta(s)$ by

$$g^\delta(s) = g(s) + \alpha \frac{\|g\|_{\Omega_2}}{\|F\|_{\Omega_2}} F(s),$$

where $F(s)$ is a smooth Chebfun function, generated by the Chebfun command `randnfun(`ϑ`,`Ω_2`)`, with maximum frequency about $2\pi/\vartheta$ and standard normal distribution $N(0, 1)$ at each point, and $\alpha > 0$ is the noise level. The noise level is defined analogously as in the discretized problems to achieve comparability. In the computed examples, we let $\vartheta = 10^{-2}$. This gives Chebfun's analogue of noise.

The discrepancy principle is used to determine the truncation parameter ℓ in (3.4) in the TSVE method, and the Tikhonov regularization parameter λ in (4.3). The discrepancy principle prescribes that the truncation index ℓ be chosen as small as possible so that the solution x_ℓ of (3.4) satisfies

$$\left\| \int_{\Omega_1} \kappa(s, t) x_\ell(t) \, dt - g^\delta(s) \right\|_{\Omega_2} \le \eta\delta,$$

where $\eta \ge 1$ is a user-supplied constant independent of δ. We let $\eta = 1$ in the examples. The discrepancy principle, when used with Tikhonov regularization, prescribes that the regularization parameter $\lambda > 0$ be chosen so that the solution x_λ of (4.1) satisfies

$$\left\| \int_{\Omega_1} \kappa(s, t) x_\lambda(t) \, dt - g^\delta(s) \right\|_{\Omega_2} = \eta\delta.$$

We use the MATLAB function `fminbnd` to find the λ-value.

---

[3] In Regularization Tools this example is called `foxgood`. We chose to name it by the full name of both authors.

Table 5.2: Comparison of TSVE with Chebfun and for the discretized problem.

Noise level	Example	n	discretized		with Chebfun	
			error	time	error	time
10^{-3}	Baart	554	$1.19 \cdot 10^{-1}$	$4.79 \cdot 10^{-2}$	$1.19 \cdot 10^{-1}$	$2.74 \cdot 10^{-1}$
	Fox-Goodwin	892	$1.44 \cdot 10^{-2}$	$8.97 \cdot 10^{-2}$	$1.43 \cdot 10^{-2}$	$1.40 \cdot 10^{-1}$
	Gravity	372	$1.97 \cdot 10^{-2}$	$2.00 \cdot 10^{-2}$	$1.97 \cdot 10^{-2}$	$3.26 \cdot 10^{-1}$
	Shaw	1694	$4.71 \cdot 10^{-2}$	$3.53 \cdot 10^{-1}$	$4.70 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$
	Wing	318	$6.03 \cdot 10^{-1}$	$1.36 \cdot 10^{-2}$	$6.03 \cdot 10^{-1}$	$2.28 \cdot 10^{-1}$
10^{-2}	Baart	226	$1.68 \cdot 10^{-1}$	$1.31 \cdot 10^{-2}$	$1.68 \cdot 10^{-1}$	$1.70 \cdot 10^{-2}$
	Fox-Goodwin	1184	$3.15 \cdot 10^{-2}$	$1.34 \cdot 10^{-1}$	$3.15 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$
	Gravity	94	$6.11 \cdot 10^{-2}$	$1.39 \cdot 10^{-3}$	$6.11 \cdot 10^{-2}$	$3.36 \cdot 10^{-2}$
	Shaw	998	$5.01 \cdot 10^{-2}$	$1.31 \cdot 10^{-1}$	$5.01 \cdot 10^{-2}$	$4.60 \cdot 10^{-2}$
	Wing	1938	$6.03 \cdot 10^{-1}$	$4.40 \cdot 10^{-1}$	$6.03 \cdot 10^{-1}$	$1.06 \cdot 10^{-2}$
10^{-1}	Baart	438	$3.45 \cdot 10^{-1}$	$5.05 \cdot 10^{-2}$	$3.45 \cdot 10^{-1}$	$3.63 \cdot 10^{-1}$
	Fox-Goodwin	1344	$5.54 \cdot 10^{-2}$	$3.69 \cdot 10^{-1}$	$5.53 \cdot 10^{-2}$	$2.94 \cdot 10^{-1}$
	Gravity	1670	$1.05 \cdot 10^{-1}$	$3.36 \cdot 10^{-1}$	$1.05 \cdot 10^{-1}$	$3.76 \cdot 10^{-1}$
	Shaw	1714	$1.37 \cdot 10^{-1}$	$3.53 \cdot 10^{-1}$	$1.35 \cdot 10^{-1}$	$7.01 \cdot 10^{-2}$
	Wing	330	$6.08 \cdot 10^{-1}$	$1.60 \cdot 10^{-2}$	$6.08 \cdot 10^{-1}$	$3.47 \cdot 10^{-1}$

One of the five test problems that we are interested in solving is `Baart`. This example is a Fredholm integral equation of the first kind (1.1) with $\kappa(s,t) = \exp(s \cos(t))$, $g(s) = 2 \sinh(s)/s$, and solution $x(t) = \sin(t)$, where $\Omega_1 = [0, \pi]$ and $\Omega_2 = [0, \pi/2]$. We compute approximate solutions by applying the TSVE and Tikhonov regularization with Chebfun. The approximate solutions, $x_\ell(t)$ and $x_\lambda(t)$, respectively, can be determined by using the formulas (3.4) and (4.3). Fig. 5.1(a) displays the kernel κ of the `Baart` example. The right-hand side function $g(s)$ and the corresponding error-contaminated function $g^\delta(s)$ are illustrated in Fig. 5.1(b), where the noise level is 10^{-2}. Fig. 5.1(c) depicts the exact solution and the computed approximate solutions determined by TSVE and Tikhonov regularization with Chebfun. The latter figure shows that our methods give fairly accurate approximations of the exact solution.

Next, we apply the methods to several different examples and compare them to standard TSVD and Tikhonov regularization in a discretized setting. The quality of the computed approximate solutions is measured by the relative error norm

$$RE := \frac{\|x_{\mathrm{method}} - x\|_*}{\|x\|_*},$$

where $\|\cdot\|_*$ denotes the scaled Euclidean vector norm $\left(\frac{1}{n} \sum_{i=1}^{n} x_i^2\right)^{1/2}$ if $\boldsymbol{x} \in \mathbb{R}^n$ is a vector, or the L^2-norm if x is a function. Tables 5.2 and 5.3 compare the TSVE and Tikhonov regularization methods when used with Chebfun to

(a) Kernel $\kappa(s,t)$

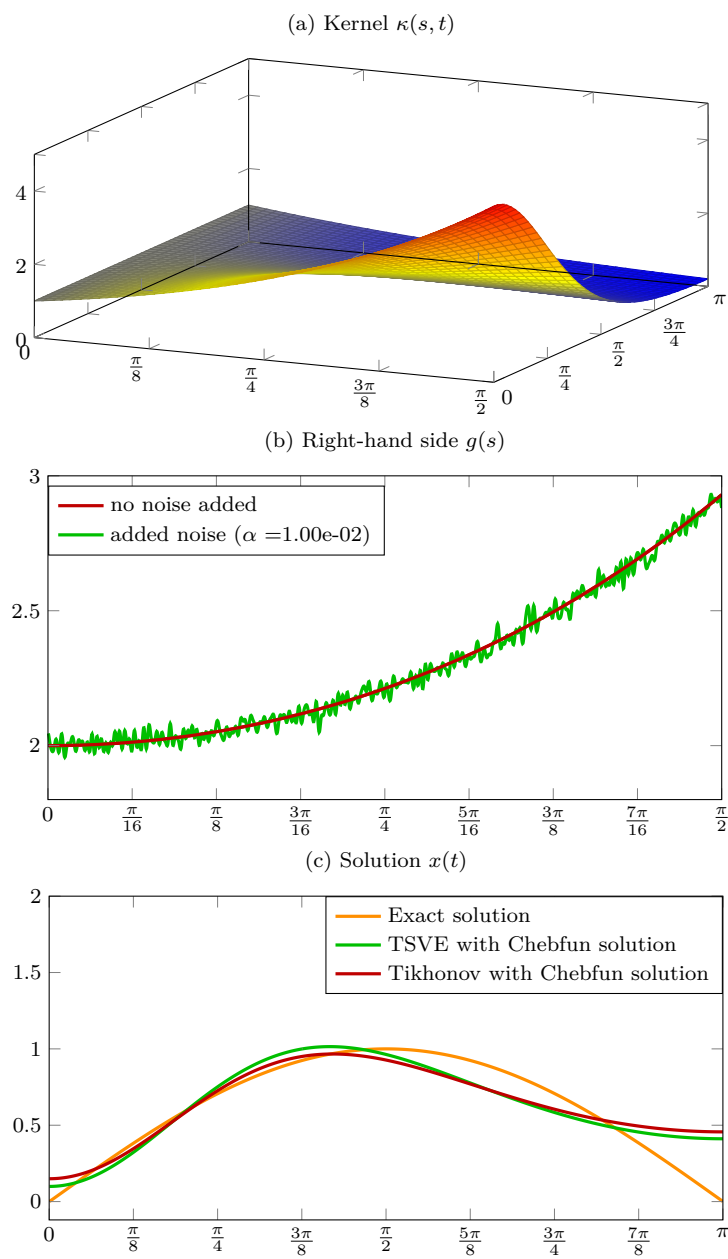
(b) Right-hand side $g(s)$

(c) Solution $x(t)$

Fig. 5.1: Example–"Baart": (a) Kernel , (b) Right-hand side, (c) Solutions.

Table 5.3: Comparison of Tikhonov with Chebfun and for the discretized problem.

Noise level	Example	n	discretized error	time	with Chebfun error	time
10^{-3}	Baart	98	$1.46 \cdot 10^{-1}$	$2.12 \cdot 10^{-3}$	$1.47 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$
	Fox-Goodwin	1484	$1.08 \cdot 10^{-2}$	$2.41 \cdot 10^{-1}$	$1.08 \cdot 10^{-2}$	$4.97 \cdot 10^{-1}$
	Gravity	214	$1.33 \cdot 10^{-2}$	$6.97 \cdot 10^{-3}$	$1.33 \cdot 10^{-2}$	$1.29 \cdot 10^{-1}$
	Shaw	562	$4.80 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$	$4.80 \cdot 10^{-2}$	$1.36 \cdot 10^{-1}$
	Wing	1672	$5.85 \cdot 10^{-1}$	$3.07 \cdot 10^{-1}$	$5.84 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$
10^{-2}	Baart	132	$1.82 \cdot 10^{-1}$	$4.72 \cdot 10^{-3}$	$1.82 \cdot 10^{-1}$	$6.48 \cdot 10^{-1}$
	Fox-Goodwin	54	$2.85 \cdot 10^{-2}$	$1.96 \cdot 10^{-3}$	$2.85 \cdot 10^{-2}$	$5.69 \cdot 10^{-1}$
	Gravity	394	$4.00 \cdot 10^{-2}$	$3.03 \cdot 10^{-2}$	$4.01 \cdot 10^{-2}$	$1.87 \cdot 10^{-1}$
	Shaw	1626	$6.25 \cdot 10^{-2}$	$3.11 \cdot 10^{-1}$	$6.25 \cdot 10^{-2}$	$1.66 \cdot 10^{-1}$
	Wing	1952	$6.03 \cdot 10^{-1}$	$4.59 \cdot 10^{-1}$	$6.03 \cdot 10^{-1}$	$1.42 \cdot 10^{-1}$
10^{-1}	Baart	270	$3.39 \cdot 10^{-1}$	$2.15 \cdot 10^{-2}$	$3.39 \cdot 10^{-1}$	$5.86 \cdot 10^{-1}$
	Fox-Goodwin	1206	$5.34 \cdot 10^{-2}$	$1.66 \cdot 10^{-1}$	$5.34 \cdot 10^{-2}$	$5.04 \cdot 10^{-1}$
	Gravity	56	$1.17 \cdot 10^{-1}$	$1.39 \cdot 10^{-3}$	$1.17 \cdot 10^{-1}$	$1.47 \cdot 10^{-1}$
	Shaw	440	$1.49 \cdot 10^{-1}$	$2.73 \cdot 10^{-2}$	$1.49 \cdot 10^{-1}$	$1.52 \cdot 10^{-1}$
	Wing	908	$6.03 \cdot 10^{-1}$	$1.11 \cdot 10^{-1}$	$6.03 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$

standard methods for the test problems `Baart`, `Fox-Goodwin`, `Gravity`, `Shaw`, and `Wing`. These problems are discretized integral equations of the first kind and are described in [13].

Three noise levels α are considered. The number of discretization points, n, which is shown in the third column of the tables, is even and chosen between 2 and 2000, such that the smallest absolute difference between the relative error of the solution for the discretized problem and the relative error of the solution for the continuous problem is achieved. Thus, we choose the number of discretization points n so that the discretized problem gives an approximate solution of about the same accuracy as the approximate solution determined with Chebfun. This choice makes a comparison of the CPU times required by the methods meaningful. The computing time required for determining this choice of n is not included in the CPU time. We remark that the determination of a suitable value of n when solving a problem of the form (2.4) can be difficult: a too large value results in unnecessarily large CPU time, while a too small value gives a computed solution of unnecessarily poor resolution. Fig. 5.2 shows the run time and accuracy for each n for the example `Shaw`. In the figure, we also annotate the different problems with the choice of the discretization. The regularize-first approach using Chebfun does not require a user to explicitly choose the discretization. This is one of the main advantages over the discretize-first approach.
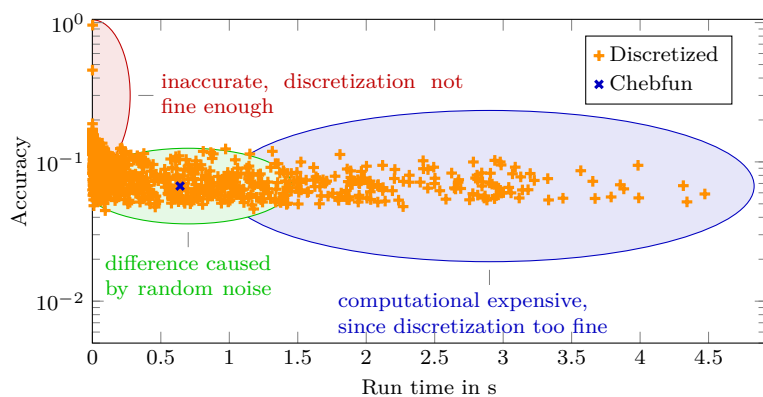
Fig. 5.2: Example—"Shaw", $\alpha = 1.00\,\mathrm{e}{-}2$.
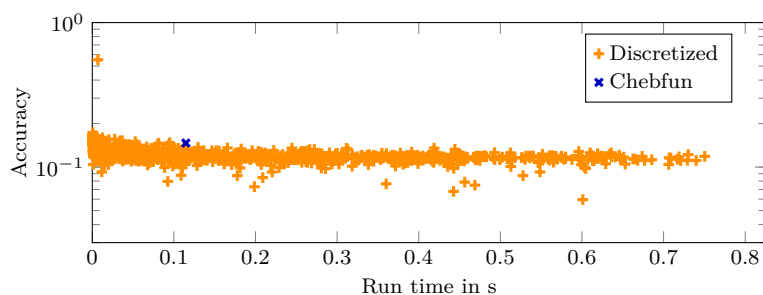
Fig. 5.3: Example –"Baart", $\alpha = 1.00\,\mathrm{e}{-}3$.

The relative errors obtained by applying TSVD and Tikhonov regularization in the discretized setting are reported in the fourth column of Tables 5.2 and 5.3, respectively. The sixth column of the tables shows the relative errors obtained when applying TSVE and Tikhonov regularization with Chebfun. We also report the CPU times in seconds for each method in the fifth and seventh columns of the tables. The tables show the computed approximate solutions determined by the Chebfun-based regularize-first methods to give as accurate approximations of the exact solutions as the approximate solutions determined by discretize-first methods. Moreover, we observe that the methods based on Chebfun are competitive with respect to run time for some problems, while they are slower for most problems. The last column of Tables 5.2 and 5.3 shows that applying TSVE with Chebfun is faster than applying Tikhonov regularization with Chebfun. This is reasonable since the TSVE method does not require the use of a root-finder.

The accuracy and run time for the discretize-first methods depend on the number of discretization points n; Chebfun-based regularize-first methods do not require a user to choose a discretization level n. Thus, in Figures 5.2–5.6, we show the relative accuracy on the vertical axis and the run time on the
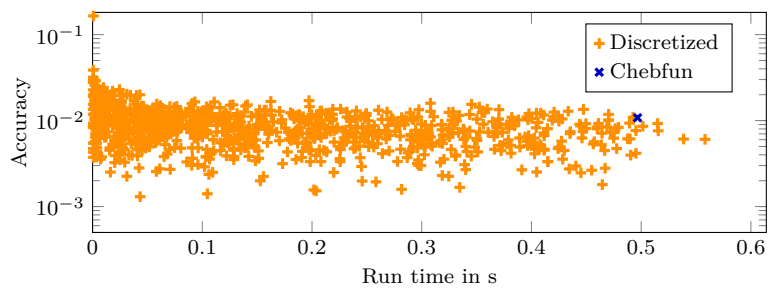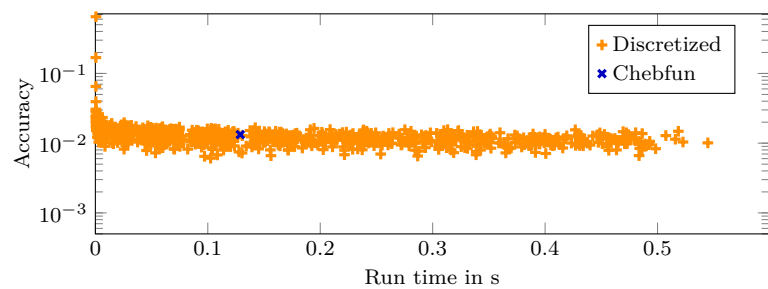
Fig. 5.4: Example –"Fox-Goodwin", $\alpha = 1.00\,\mathrm{e}{-3}$.

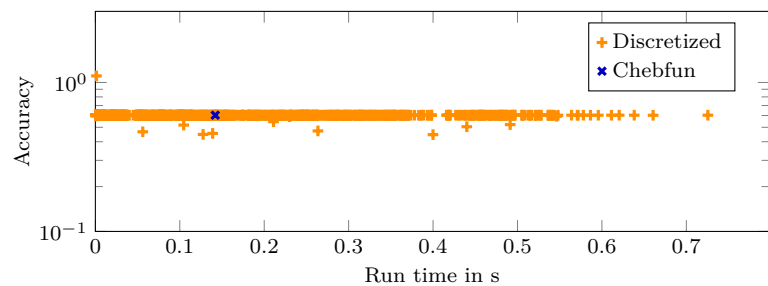Fig. 5.5: Example –"Gravity", $\alpha = 1.00\,\mathrm{e}{-3}$.

Fig. 5.6: Example –"Wing", $\alpha = 1.00\,\mathrm{e}{-2}$.

horizontal axis; being closer to the origin is better. The figures illustrate that the accuracy and computing time of the implementations with Chebfun are competitive.

6 Problems in two space-dimensions

Linear ill-posed problems based on integral equations in one space-dimension are arguably less challenging than ill-posed problems in two space-dimensions. We therefore also consider Fredholm integral equations of the first kind in two

space-dimensions,

$$\int_{\Omega_1} \kappa(s_1, s_2, t_1, t_2) x(t_1, t_2)\, dt_1 dt_2 = g^\delta(s_1, s_2), \qquad (s_1, s_2) \in \Omega_2. \qquad (6.1)$$

The obvious main difference from the situation discussed above is that κ is now a function of four variables. This poses a main obstacle, since Chebfun does not have a Chebfun4 version for the computation with functions of four variables. We will discuss two different examples, Gaussian `blur` and inverse `diffusion`, that illustrate how this limitation can be overcome. The solution methods for these problems differ, since `blur` has a kernel that can be written as a product and we will exploit this. Solving `diffusion` requires a more general, but also more expensive, approach.

6.1 Gaussian blur

The problem `blur` models Gaussian blur with a kernel that is based on a tensor product of 1D-blur kernels. It is given by

$$\kappa(s_1, s_2, t_1, t_2) = \kappa_1(s_1, t_1) \times \kappa_2(s_2, t_2) \qquad (6.2)$$

with

$$\kappa_i(s_i, t_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t_i - s_i)^2}{2\sigma^2}\right), \quad i = 1, 2,$$

where σ is the standard deviation of the Gaussian distribution. The exact solution $x(t_1, t_2)$ is constructed as a continuous "image"[4] that we blur and contaminate by noise, and then try to reconstruct. In our example, we let $\sigma = 0.2$ and construct the exact solution as

$$x(t_1, t_2) = \{(t_1, t_2) \in \Omega : -0.5 \le t_1 \le 0.2 \text{ and } -0.6 \le t_2 \le -0.2\},$$

which is shown in Fig. 6.1(a).[5] The error-free right-hand side function is determined by

$$g(s_1, s_2) := \int_\Omega \kappa(s_1, s_2, t_1, t_2)\, x(t_1, t_2)\, dt_1 dt_2$$

and the error-contaminated function $g^\delta(s_1, s_2)$ in (6.1) is defined by

$$g^\delta(s_1, s_2) = g(s_1, s_2) + \alpha \frac{\|g\|_{\Omega_2}}{\|F\|_{\Omega_2}} F(s_1, s_2),$$

---

[4]With a continuous "image" we mean a mapping from $[0, 1] \times [0, 1]$ to $[0, 1]$, where the function value represents a gray-scale value. Thus, a gray-scale value exists for all points continuously and not just for discrete points on a grid. The mapping itself is not necessarily continuous.

[5]Note that for the plot we, as it is customary, evaluated the function on a grid. The plot shows linear interpolation of the function values at the grid points.

where $F(s_1, s_2)$ is a smooth Chebfun function in two space-dimensions with maximum frequency about $2\pi/\vartheta$ and standard normal distribution $N(0,1)$ at each point; α is the noise level. In this problem, we let the noise level and ϑ equal 10^{-2}. The error-free right-hand side is shown in Fig. 6.1(c), and Fig. 6.1(e) depicts the noise that we added to generate the error-contaminated function g^δ.

A kernel that can be separated into a product of two functions as in (6.2) can be handled by Chebfun. We first compute the singular value expansions of κ_1 and κ_2. They allow us to approximate κ as

$$\kappa(s_1, s_2, t_1, t_2) \approx \sum_{i=1}^{r_1} \sigma_i \phi_i^{(1)}(s_1) \psi_i^{(1)}(t_1) \sum_{j=1}^{r_2} \mu_j \phi_j^{(2)}(s_2) \psi_j^{(2)}(t_2), \qquad (6.3)$$

where both the σ_i and μ_j denote singular values. We seek an approximate solution in the form

$$x(t_1, t_2) = \sum_{k=1}^{r_1} \sum_{\ell=1}^{r_2} \beta_{k\ell} \psi_k^{(1)}(t_1) \psi_\ell^{(2)}(t_2). \qquad (6.4)$$

By substituting (6.3) and (6.4) into (6.1), and using the orthonormality of the basis functions, we get

$$\sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sigma_i \mu_j \beta_{ij} \phi_i^{(1)}(s_1) \phi_j^{(2)}(s_2) = g^\delta(s_1, s_2).$$

We probe this equation with $\phi_p^{(1)}(s_1) \phi_q^{(2)}(s_2)$ for all p and q, and use the orthonormality of the basis functions, to obtain

$$\beta_{ij} = \frac{\int_{\Omega_2} g^\delta(s_1, s_2) \phi_i^{(1)}(s_1) \phi_j^{(2)}(s_2) \, ds_1 ds_2}{\sigma_i \mu_j}. \qquad (6.5)$$

This allows us to implement the solution algorithm using functions of at most two variables and, thus, do not exceed the capabilities of Chebfun2. For the solution of the forward problem, that is for blurring the exact solution in the setup process of the example, we use Chebfun3 [14],

$$g(s_1, s_2) = \int_{\Omega_1} \kappa(s_1, s_2, t_1, t_2) \, x(t_1, t_2) \, dt_1 dt_2$$

$$= \int_{\Omega_{1,2}} \int_{\Omega_{1,1}} \kappa_1(s_1, t_1) \, x(t_1, t_2) \, dt_1 \, \kappa_2(s_2, t_2) \, dt_2.$$

The product $\kappa_1(s_1, t_1) \, x(t_1, t_2)$ is a function of three variables, s_1, t_1, and t_2. The result of the inner integral is a function of s_1 and t_2. Thus, the product with $\kappa_2(s_2, t_2)$ is another function of three variables.

Equation (6.3) provides a singular function expansion that also can be used for Tikhonov regularization. Instead of solving (6.1) exactly, we solve

$$\min_{x \in L^2(\Omega_1)} \left\{ \left\| \int_{\Omega_1} \kappa(\cdot, \cdot, t_1, t_2) x(t_1, t_2) \, dt_1 \, dt_2 - g^\delta \right\|^2_{\Omega_2} + \lambda^2 \|x\|^2_{\Omega_1} \right\}. \qquad (6.6)$$

Using the fact that the kernel is separable, substituting (6.3) and (6.4) into (6.6), and using the orthonormality of the basis functions, we get

$$\min_{\beta_{ij}} \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \left(\beta_{ij}^2 \sigma_i^2 \mu_j^2 - 2\beta_{ij}\sigma_i\mu_j \int_{\Omega_2} \phi_i^{(1)}(s_1) \phi_j^{(2)}(s_2) \; g^\delta(s_1, s_2) \, ds_1 \, ds_2 + $$
$$\lambda^2 \beta_{ij}^2 \right) + \int_{\Omega_2} \left| g^\delta(s_1, s_2) \right|^2 ds_1 \, ds_2 \qquad (6.7)$$

and obtain the solution

$$x_\lambda(t_1, t_2) = \sum_{k=1}^{r_1} \sum_{\ell=1}^{r_2} \beta_{k\ell} \psi_k^{(1)}(t_1) \psi_\ell^{(2)}(t_2),$$

$$\text{with } \beta_{k\ell} = \frac{\sigma_k \mu_\ell \int_{\Omega_2} \phi_k^{(1)}(s_1) \phi_\ell^{(2)}(s_2) \; g^\delta(s_1, s_2) \, ds_1 \, ds_2}{\sigma_k^2 \mu_\ell^2 + \lambda^2}. \qquad (6.8)$$

This provides us with the tools to reconstruct the exact image $x(t_1, t_2)$ from its blurred version. Similarly as for the problems in one space-dimension, the truncation parameter ℓ in (6.4) and the Tikhonov regularization parameter λ in (6.8) are determined with the aid of the discrepancy principle, where we set η to be 10 in our example. The reconstructed images obtained with Tikhonov and TSVE regularization with Chebfun are shown in Figures 6.1(b) and (d), respectively. The two reconstructed images are seen to be of roughly the same quality, with the image determined by Tikhonov regularization being slightly less oscillatory. The computing times for both methods differ significantly: the TSVE with Chebfun required 411 seconds, while Tikhonov regularization with Chebfun took 1206 seconds.

6.2 Inverse diffusion

We next consider an integral equation with a non-separable kernel $\kappa(s_1, s_2, t_1, t_2)$. In the following, we will first describe the example and then discuss a method to solve the problem. This example is a continuous version of the `PRdiffusion` example provided in IR Tools [10].

The partial differential equation

$$\dot{u} = u_{t_1, t_1} + u_{t_2, t_2}, \qquad \text{for all } (t_1, t_2) \in \Omega_1,$$

describes a diffusion process, where $u(t_1, t_2, \tau)$ is the concentration at the point (t_1, t_2) in Ω_1 at time τ. The time derivative of u is denoted by $\dot{u}$, and u_{t_i, t_i}

(a) Exact solution $x(t_1, t_2)$.

(b) Computed solution $x_\lambda(t_1, t_2)$

(c) Right-hand side $g(s_1, s_2)$

(d) Computed solution $x_k(t_1, t_2)$

(e) Noise added to $g(s_1, s_2)$

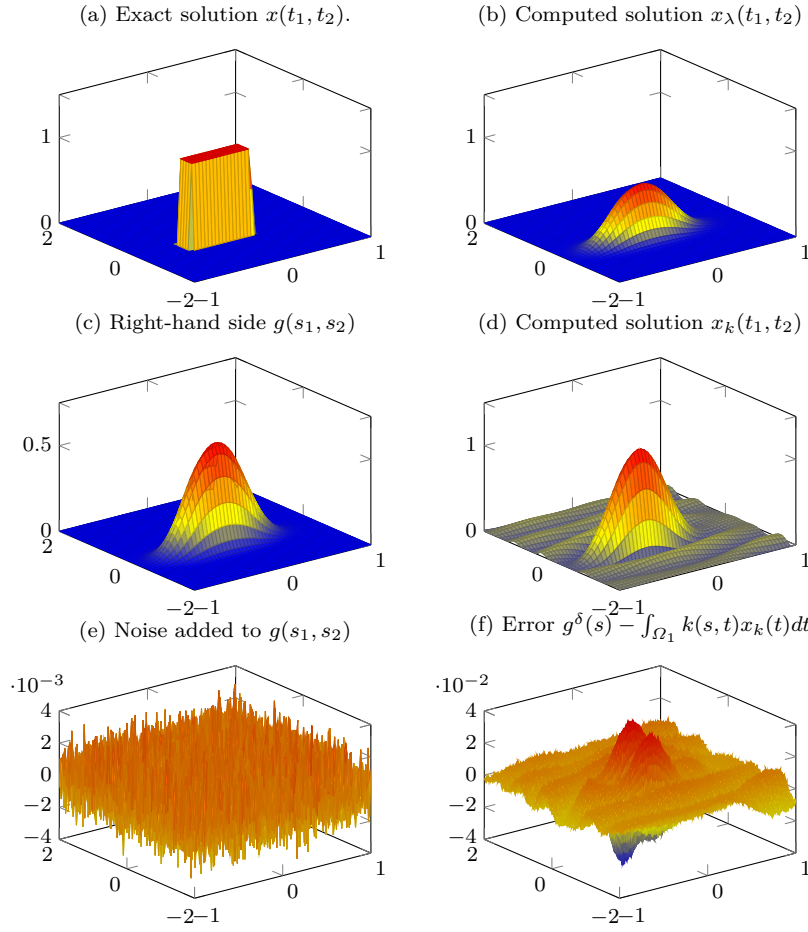(f) Error $g^\delta(s) - \int_{\Omega_1} k(s, t) x_k(t) dt$

Fig. 6.1: Example –"Blur2D", $\alpha = 1.00\,e{-}2$.

stands for the second derivative in direction t_i. We assume that u satisfies the initial condition

$$u(t_1, t_2, 0) = u_0(t_1, t_2),$$

for some given function u_0, and Neumann boundary conditions for all time $\tau \geq 0$,

$$u_{t_1}(t_1, t_2, \tau) = 0, \qquad \text{for all } (t_1, t_2) \in \partial\Omega_1,$$

where $\partial\Omega_1$ denotes the boundary of Ω_1. After T seconds, the solution of the system is

$$u_T(t_1, t_2) = u(t_1, t_2, T).$$

We assume that u_T or a noisy version of u_T are given, and our task is to recover u_0. In this example, we have $\Omega_1 = [0, 1] \times [0, 1]$ and $T = 0.01$. The

initial condition is

$$u_0(t_1, t_2) = 0.7 \exp\left(-\left(\tfrac{t_1-0.4}{0.12}\right)^2 - \left(\tfrac{t_2-0.5}{0.15}\right)^2\right) + \exp\left(-\left(\tfrac{t_1-0.7}{0.1}\right)^2 - \left(\tfrac{t_2-0.4}{0.08}\right)^2\right).$$

The fundamental solution of this problem is given by

$$u(t_1, t_2, \tau) = \int_0^1 \int_0^1 \Big(\Gamma(t_1 - s_1, \tau)\Gamma(t_2 - s_2, \tau) +$$
$$\Gamma(t_1 + s_1, \tau)\Gamma(t_2 + s_2, \tau)\Big)u_0(s_1, s_2)\, ds_1\, ds_2,$$

where

$$\Gamma(x, \tau) = \frac{1}{\sqrt{4\pi\tau}} \exp\left(\frac{-\tau^2}{4t}\right).$$

We have

$$\int_0^1 \int_0^1 \kappa(s_1, s_2, t_1, t_2)u_0(s_1, s_2)\, ds_1\, ds_2 = u_T(t_1, t_2).$$

The right-hand side is computed by solving the forward problem. As in the previous examples, we use Chebfun's random function `randnfun2` to generate two-dimensional noise; see Fig. 6.2(e). The kernel κ can be written as

$$\kappa(s_1, s_2, t_1, t_2) = \kappa_1(s_1, t_1)\kappa_2(s_2, t_2) + \kappa_3(s_1, t_1)\kappa_4(s_2, t_2). \tag{6.9}$$

This kernel has a special structure for which one could construct a specific method. Instead, we opted for implementing a more general method. As in the one-dimensional case, we use cross approximation followed by a singular value decomposition to compute the singular value expansion of κ. This method follows the description in Subsection 2.1 with the difference that $s = (s_1, s_2)$ and $t = (t_1, t_2)$ here. To find an approximation to the maximum κ in $\Omega_2 \times \Omega_1$, we use the MATLAB function `fminsearch`. We choose as pivot for the cross approximation the maximizer $(\hat{s}_1, \hat{s}_2, \hat{t}_1, \hat{t}_2)$. After computing the crosses $C_1(s) = \kappa(s_1, s_2, \hat{t}_1, \hat{t}_2)$ and $R_1(t) = \kappa(\hat{s}_1, \hat{s}_2, t_1, t_2)$, we update the remainder $\kappa - C_1(s)M_1R_1(t)^T$. We then repeat this process for the remainder so-obtained until the maximum is below a preset tolerance. The columns of $C(s)$ and rows of $R(t)$ determined by cross approximation are chebfun2 objects. Nevertheless, orthonormalization based on the Gram–Schmidt process is straightforward using Chebfun2. Thereby, one also obtains upper triangular matrices R_c and R_r, and the remaining computations are identical to those described in Subsection 2.1.

We developed a proof-of-concept implementation of this method. Our implementation lacks the run time optimization that a possible future Chebfun4 extension provided by the Chebfun developers would have.

Tikhonov regularization determines the solution f_λ, which is shown in Fig. 6.2(b); the exact solution is displayed in Fig. 6.2(a) for comparison. Since our implementation of Tikhonov regularization uses the singular value expansion, we can easily solve the problem using TSVE, too. The TSVE regularized

solution is shown in Fig. 6.2(d) and is visually about the same as the solution determined by Tikhonov regularization, despite having a slightly higher relative norm-wise error. Fig. 6.2(c) shows the exact right-hand side, Fig. 6.2(e) depicts the added noise, and the difference between the right-hand side computed from the TSVE regularized solution and the right-hand side with noise g^δ is shown in Fig. 6.2(f).

Our proof-of-concept implementation required about half an hour for the computations including about 20 minutes for the singular value expansion and about 10 minutes for determining λ using the discrepancy principle.

This method works for all kernel functions of four variables. However, the fact that κ can be written as in (6.9) ensures that the columns of $C(s)$ and the rows of $R(t)$ can each be represented by chebfun2 objects of rank 2. The ranks are relevant for the run time of the method. When applying the Gram–Schmidt process and when computing the singular functions, the ranks increase and are automatically truncated by Chebfun's arithmetic. Nevertheless, the fact that the rank initially is 2 has a dampening effect on the computational complexity.

7 Conclusion

The computed results illustrate the feasibility of using Chebfun to solve linear discrete ill-posed problems and in this way carry out computations in a fashion that is close to the spirit of the analysis of ill-posed problems found, e.g., in [7]. The accuracy and timings of the implementations with Chebfun are competitive.

In the future, further extensions to Chebfun including the treatment of functions of four or six variables will allow the application of the Chebfun-based approach discussed in this paper to the solution of linear ill-posed problems in two and three space-dimensions. In the meantime, the two methods described in Section 6 can be used.

(a) Exact solution $x(t_1, t_2)$.

(b) Computed solution $x_\lambda(t_1, t_2)$

(c) Right hand side $g(s_1, s_2)$

(d) Computed solution $x_k(t_1, t_2)$

(e) Noise added to $g(s_1, s_2)$

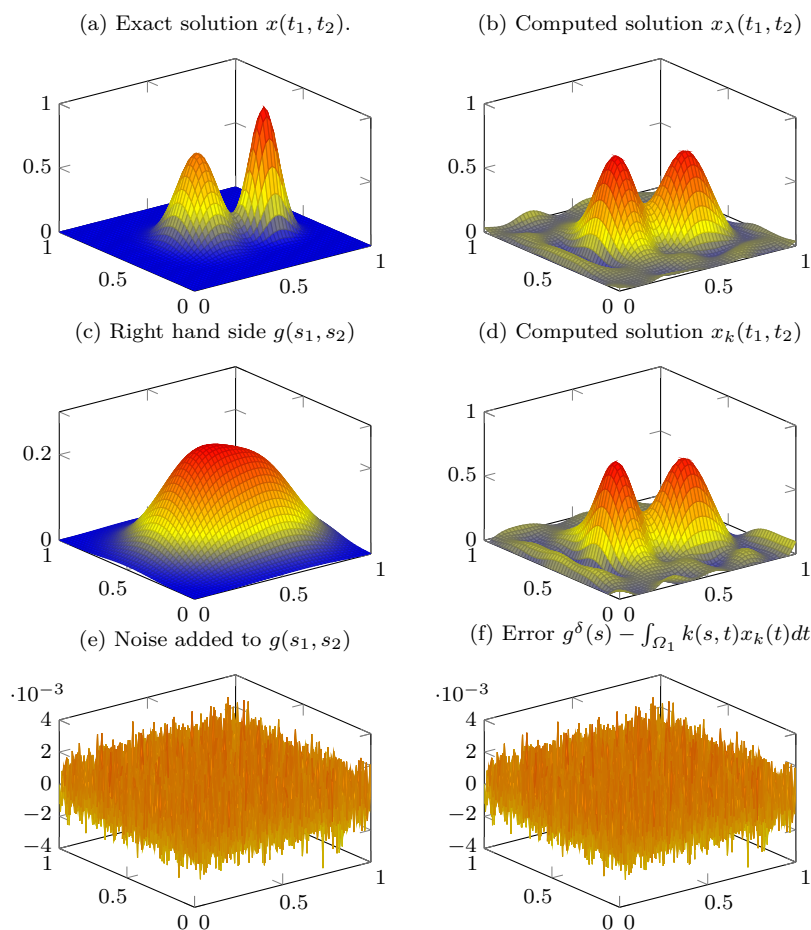(f) Error $g^\delta(s) - \int_{\Omega_1} k(s,t) x_k(t) dt$

Fig. 6.2: Example—"Diffusion", $\alpha = 1.00\,\mathrm{e}{-2}$.

References

1. M. L. BAART, *The use of auto-correlation for pseudo-rank determination in noisy ill-conditioned linear least-squares problems*, IMA Journal of Numerical Analysis, 2 (1982), pp. 241–247.

2. M. BEBENDORF, *Approximation of boundary element matrices*, Numerische Mathematik, 86 (2000), pp. 565–589.

3. M. BEBENDORF AND S. RJASANOW, *Adaptive low-rank approximation of collocation matrices*, Computing, 70 (2003), pp. 1–24.

4. C. BREZINSKI, G. RODRIGUEZ, AND S. SEATZU, *Error estimates for linear systems with applications to regularization*, Numerical Algorithms, 49 (2008), pp. 85–104.

5. D. CALVETTI AND L. REICHEL, *Tikhonov regularization of large linear problems*, BIT Numerical Mathematics, 43 (2003), pp. 263–283.

6. T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, eds., *Chebfun Guide*, Oxford, 2014.

7. H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 2000.

8. C. Fenu, L. Reichel, and G. Rodriguez, *GCV for Tikhonov regularization via global Golub–Kahan decomposition*, Numerical Linear Algebra with Applications, 23 (2016), pp. 467–484.

9. L. Fox and E. T. Goodwin, *The numerical solution of non-singular linear integral equations*, Philosophical transactions of the Royal Society of London. Series A Mathematical, Physical and Engineering Sciences, 245 (1953), pp. 501–534.

10. S. Gazzola, P. C. Hansen, and J. G. Nagy, *IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems*, Numerical Algorithms, 81 (2019), pp. 773–811.

11. W. Hackbusch, *Integral Equations: Theory and Numerical Treatment*, International Series on Numerical Mathematics, Birkhäuser, 1995.

12. P. C. Hansen, *Rank-definicient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

13. P. C. Hansen, *Regularization tools version 4.0 for Matlab 7.3*, Numerical Algorithms, 46 (2007), pp. 189–194.

14. B. Hashemi and L. N. Trefethen, *Chebfun in three dimensions*, SIAM Journal on Scientific Computing, 39 (2017), pp. C341–C363.

15. S. Kindermann, *Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems*, Electronic Transactions on Numerical Analysis, 38 (2011), pp. 233–257.

16. S. Kindermann and K. Raik, *A simplified L-curve method as error estimator*, Electronic Transactions on Numerical Analysis, 53 (2020), pp. 217–238.

17. T. Mach, L. Reichel, M. Van Barel, and R. Vandebril, *Adaptive cross approximation for ill-posed problems*, Journal of Computational and Applied Mathematics, 303 (2016), pp. 206–217.

18. F. Natterer, *Regularization of ill-posed problems by projection methods*, Numerische Mathematik, 28 (1977), pp. 329–341.

19. L. Reichel and G. Rodriguez, *Old and new parameter choice rules for discrete ill-posed problems*, Numerical Algorithms, 63 (2013), pp. 65–87.

20. L. Reichel, G. Rodriguez, and S. Seatzu, *Error estimates for large-scale ill-posed problems*, Numerical Algorithms, 51 (2009), pp. 341–361.

21. E. Schmidt, *Zur Theorie der linearen und nichtlinearen Integralgleichungen*, Vieweg+Teubner Verlag, Leipzig/Wiesbaden, 1989, pp. 190–233; reprint of an article from 1905.

22. C. B. Shaw Jr., *Improvement of the resolution of an instrument by numerical solution of an integral equation*, Journal of Mathematical Analysis and Applications, 37 (1972), pp. 83–112.

23. A. Townsend and L. N. Trefethen, *An extension of Chebfun to two dimensions*, SIAM Journal on Scientific Computing, 35 (2013), pp. C495–C518.

24. G. M. Wing, *A primer on integral equations of the first kind: the problem of deconvolution and unfolding*, SIAM, 1991.

25. N. Yarvin and V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators*, SIAM Journal on Scientific Computing, 20 (1998), pp. 699–718.