Numerical Identification of Coefficient for Parabolic Equation

D. Kh. Ivanov i.am.djoos@gmail.com
Ammosov North-Eastern Federal University

Introduction

Inverse problems for finding the coefficients of ODEs take place since very often in applied problems it is necessary to determine the properties of investigated medium (thermal conductivity, resistance, geometric parameters etc.), which enter into coefficient parts of ODEs. However, in many cases these properties are difficult to calculate directly, but thanks to additional information, when the task becomes well-posed, they can be restored. Among such problems, we single out one where it is necessary to restore the diffusion coefficient of the parabolic equation in the two-dimensional domain. We also consider this coefficient depends on only the solution itself, in [1] and other works one usually takes it with space and time dependencies. As additional information, one can take observations of the solution in inner points of the domain through whole time period. In this work, the numerical identification of the diffusion coefficient in the parabolic equation by the finite element method performed by the FEniCS package and the *dolfin_adjoint* package [2].

Forward problem

Define domain
$\boldsymbol{x} = (x_1, x_2) \in \Omega = (0,1) \times (0,1), \quad \partial\overline{\Omega} = \Gamma_D \cup \Gamma_N,$
$\Gamma_D = \{(x_1, x_2) : x_1 \in \{0,1\}\}, \quad \Gamma_N = \{(x_1, x_2) : x_2 \in \{0,1\}\}.$

Consider the next boundary problem

$$\frac{\partial u}{\partial t} - \sum_{k=1}^{2} \frac{\partial}{\partial x_k}\left(k(u)\frac{\partial u}{\partial x_k}\right) = f(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega, \quad 0 < t \le T,$$

$$u(\boldsymbol{x}, t) = g(t), \quad \boldsymbol{x} \in \Gamma_D, \quad 0 < t \le T,$$

$$\frac{\partial u}{\partial n}(\boldsymbol{x}, t) = 0, \quad \boldsymbol{x} \in \Gamma_N, \quad 0 < t \le T,$$

$$u(\boldsymbol{x}, 0) = 0, \quad \boldsymbol{x} \in \Omega.$$

Coefficient inverse problem

Find $k(u)$ with additional condition

$$u(\boldsymbol{x}_m, t) = d_m(t), \quad m = 1, 2, ..., M.$$

Define control $\boldsymbol{c}$ as a weight-vector for $k(u)$ decomposition

$$k(u) = \sum_{n=1}^{N} c_n \varphi_n(u).$$

Piecewise-constant basis:

$$\varphi_n(u) = \begin{cases} 1, & if \quad u \in [u_{n-1}, u_n], \\ 0, & else. \end{cases}$$

Piecewise-linear basis:

$$\varphi_n(u) = \begin{cases} \dfrac{u - u_{n-1}}{u_n - u_{n-1}}, & if \quad u \in [u_{n-1}, u_n], \\ \dfrac{u_{n+1} - u}{u_{n+1} - u_n}, & if \quad u \in [u_n, u_{n+1}], \\ 0, & else. \end{cases}$$

Variational problem

Define residual functional as following [3, 4]

$$J(\boldsymbol{c}) = \sum_{m=1}^{M} \int_0^T \int_\Omega \delta(\boldsymbol{x} - \boldsymbol{x}_m)(u(\boldsymbol{x}, t; \boldsymbol{c}) - d_m(t))^2 d\boldsymbol{x}dt.$$

Numerical solution for the coefficient satisfies

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} J(\boldsymbol{c}).$$

Algorithm

1. Define $\mathcal{D}$ as a thrust interval for k and divide it into N subintervals $\omega_n = [u_{n-1}, u_n]$
2. Choose basis function $\varphi_n(u)$
3. Initialize $\boldsymbol{c} = \boldsymbol{c}^j$, $j = 0$ and ε
4. Solve forward problem considering the from for $k(u; \boldsymbol{c}^j)$
5. Find $\nabla J(\boldsymbol{c}^j)$
6. If $\max_n |\nabla_n J(c^j)| < \varepsilon$ then exit, else go to the next step
7. $\boldsymbol{c}^{j+1} = \boldsymbol{c}^j + \Delta, \quad \Delta$ is defined by SciPy optimization method
8. $j = j + 1$ and go to the step 4

Model problem

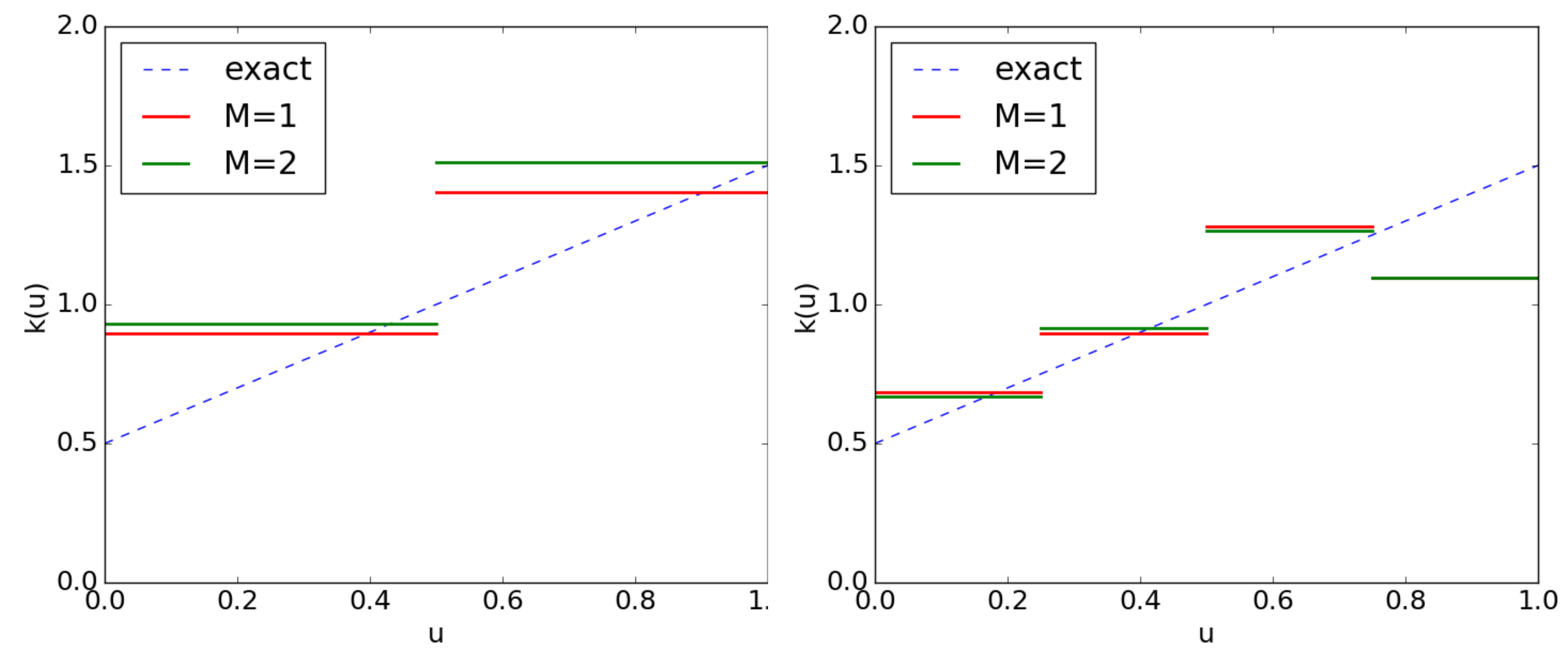
$$f(\boldsymbol{x}, t) = 0,$$

$$k(u) = 0.5 + u \quad \text{— exact solution},$$

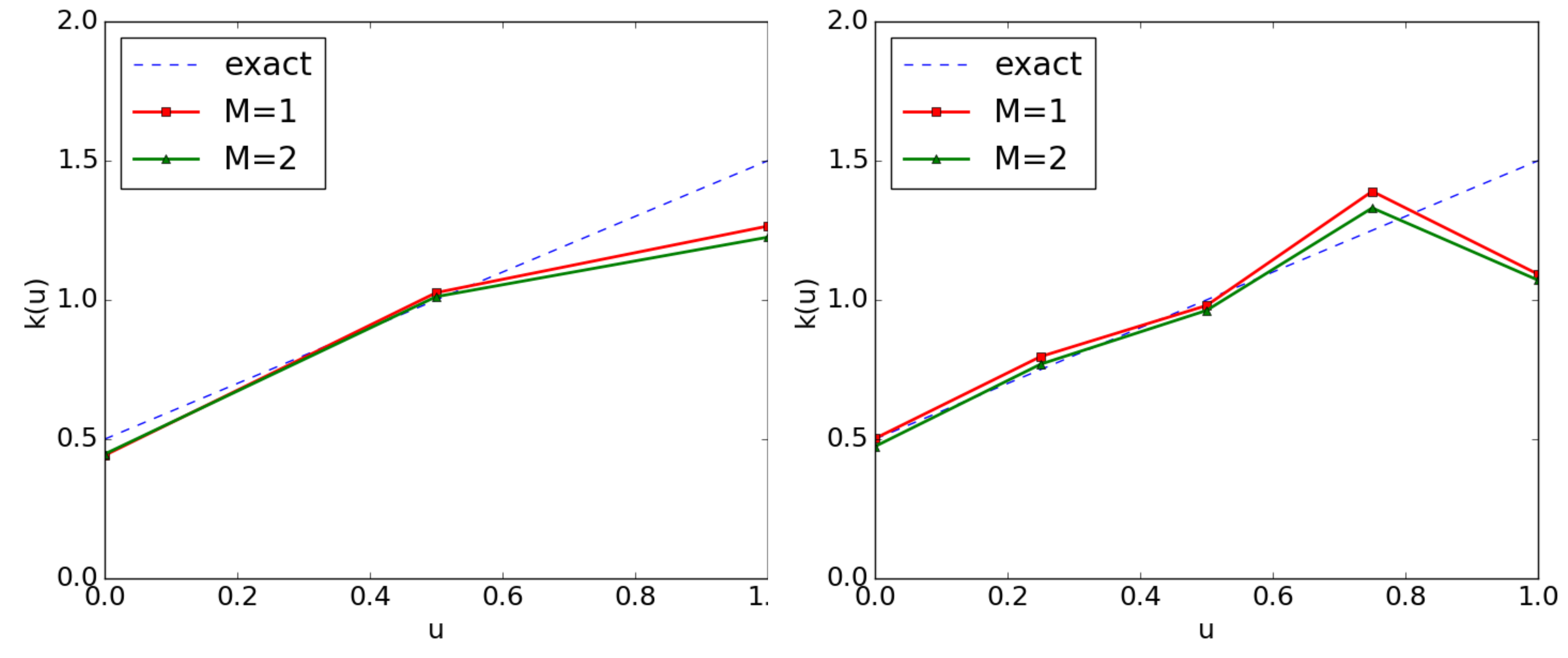
$$g(t) = \begin{cases} 0, & if \quad x_1 = 0, \\ t, & if \quad x_1 = 1. \end{cases}$$

Results

Piecewise-constant basis



Piecewise-linear basis



Future research

Further investigate the problem, particularly,
- adding regularization,
- considering different model problems whith different exact coefficient.

References

[1] Alemdar Hasanov, Paul DuChateau, and B Pektaş. An adjoint problem approach and coarse-fine mesh method for identification of the diffusion coefficient in a linear parabolic equation. *Journal of Inverse and Ill-posed Problems jiip*, 14(5):435–463, 2006.

[2] Patrick E Farrell, David A Ham, Simon W Funke, and Marie E Rognes. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing*, 35(4):C369–C393, 2013.

[3] OM Alifanov, EA Artyukhin, and SV Rumyantsev. *Extremum Methods to Solve Ill-Posed Problems {in Russian}*. Nauka, Moscow, 1988.

[4] Fedor Pavlovich Vasil'ev. *Chislennye metody resheniia ekstremal'nykh zadach {in Russian}*. Nauka, Moscow, 1980.