Journal of Neurodevelopmental Cognition

Journal of Neurodevelopmental Cognition 1 (2022) 51-57
ISSN: 2645-565X
http://www.jncog.sbu.ac.ir

10.29252/JNCOG.2022.103422

A computational source modeling of brain activity: An inverse problem

Amir Hosein Hadian Rasanan[a], Jamal Amani Rad[b,*]

[a]*Department of Computer Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran*
[b]*Department of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University, Tehran, Iran*

Abstract

Source determination in an inverse problem from the over-specified data plays a crucial role in cognitive modeling. In this paper, an accurate and fast method is proposed for solving the one-dimensional inverse problem concerning diffusion equation with source control parameter. The proposed method is based on applying a compact finite difference scheme for spatial components and solving the system that arised from this scheme by multigrid.

Keywords: Brain activity, Inverse problem, Control parameter, Computational modeling.

1. Introduction

Source determination in an inverse problem from the over-specified data plays a crucial role in several physical phenomena. This technique has been widely used to determine the unknown properties of a region by measuring data only on its boundary or a specified location in the domain. These unknown properties, such as the neuronal activity sources corresponding to a set of measured data (electric potential or magnetic fields), are important to obtain information on the brain activity, but they usually cannot be measured directly, or the process of their measurement is very expensive. For the sake of simplicity exposition, we restrict our attention to an one-dimensional inverse model, but the case of high-dimensional models can be treated in perfect analogy. So, in this paper, we shall consider an inverse problem of finding a source parameter $p(t)$ in the following diffusion equation:

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} + p(t)w + \phi(x,t), \quad 0 \leq x \leq 1, \quad 0 < t \leq T, \tag{1.1}$$

*Corresponding author
Email addresses: amir.h.hadian@gmail.com (Amir Hosein Hadian Rasanan), j_amanirad@sbu.ac.ir (Jamal Amani Rad)

Received: March 2022 *Revised:* May 2022

with the initial condition

$$w(x,0) = f(x), \quad 0 \le x \le 1, \tag{1.2}$$

and the boundary conditions

$$w(0,t) = g_0(t), w(1,t) = g_1(t) \quad 0 < t \le T, \tag{1.3}$$

subject to the over specification over a portion of the spatial domain

$$\int_0^{s(t)} w(x,t)dx = E(t), \quad 0 < s(t) \le 1, \quad 0 < t \le T, \tag{1.4}$$

where f, g_0, g_1, s, ϕ and E are the suitable known functions. While the functions w and p are unknown. By employing a pair of transformations $r(t) = e^{-\int_0^t p(s)ds}$ and $u(x,t) = r(t)w(x,t)$, we have $p(t) = \frac{-r'(t)}{r(t)}$. So, an alternative for Eqs. (1)-(4) related to the previous relation, can be mentioned as

$$u_t = u_{xx} + r(t)\phi(x,t), \tag{1.5}$$

subject to

$$u(x,0) = f(x), \quad 0 \le x \le 1, \tag{1.6}$$

$$u(0,t) = r(t)g_0(t) = q_0(t), \quad 0 < t \le T, \tag{1.7}$$

$$u(1,t) = r(t)g_1(t) = q_1(t), \quad 0 < t \le T, \tag{1.8}$$

and

$$\int_0^{s(t)} u(x,t)dx = r(t)E(t), \tag{1.9}$$

or, equivalently,

$$r(t) = \frac{\int_0^{s(t)} u(x,t)dx}{E(t)}. \tag{1.10}$$

It is important to observe that in relations (5)-(10) exist unknown functions, so we should approximate these functions. To this aim coupling the compact finite difference scheme and multigrid algorithm will be used for spatial and time approximations, as described in the next section.

2. Methodology

Before we show how to discretize model in the form (5)-(10), we focus on the numerical tools which are used in this work. First of all , the spacial derivative is discrerized by a compact finite difference which is four-order accurate as follows:

$$(u_{xx})_{i-1} + 10(u_{xx})_i + (u_{xx})_{i+1} = \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}) + O(h^4), \tag{2.1}$$

Second, a crucial point in proposed method is an accurate evaluation of the integrals equation (10). Since the integral part is nonlocal and highly complicated, an accurate numerical integration is highly difficult. In this work, we consider the following numerical integration: we decompose it into two separate parts, for approximating first part we have employed $O(h^5)$ Simpson's composite rule and for approximating second part we have applied quadratic interpolating:

$$\int_0^{s(t)} u(x,t)dx = \int_0^{3lh} u(x,t)dx + \int_{3lh}^{s(t)} u(x,t)dx, \tag{2.2}$$

where $l = \lfloor \frac{s(t)}{3h} \rfloor$, and substituting in the second integral of $z = \frac{x}{h} - 3l$ yields

$$\int_{3lh}^{s(t)} u(x,t)dx = h\int_0^{\gamma(t)} u(z,t)dz, \tag{2.3}$$

where $\gamma(t) = \frac{s(t)}{h} - 3l$.
Replacement of u in the integral with a quadratic interpolating polynomial (the Newton's for-ward-difference formula) through the grid values concerned, gives:

$$\begin{aligned} h\int_0^{\gamma(t)} u(z,t)dz = h\int_0^{\gamma(t)} \Big[& u(z_{3l},t) + z\Delta u(z_{3l},t) + \frac{1}{2}z(z-1)\Delta^2 u(z_{3l},t) \\ & + \frac{1}{6}z(z-1)(z-2)\Delta^3 u(z_{3l},t)\Big]dz, \end{aligned} \tag{2.4}$$

where : $\Delta u(z_{3l},t) = u(z_{3l+1},t) - u(z_{3l},t)$ and $\Delta^2 u(z_{3l},t) = u(z_{3l+2},t) - 2u(z_{3l+1},t) + u(z_{3l},t)$ also $\Delta^3 u(z_{3l},t) = u(z_{3l+3},t) - 3u(z_{3l+2},t) + 3u(z_{3l+1},t) - u(z_{3l},t)$
By applying above procedure we obtain:

$$\begin{aligned} \int_0^{s(t)} u(x,t)dx = {} & \frac{3h}{8}\Big(u(z_0,t) + 3\sum_{i=0}^{l-1} u(z_{3i+1},t) + 3\sum_{i=0}^{l-1} u(z_{3i+2},t) + 2\sum_{i=1}^{l-1} u(z_{3i},t) + u(z_{3l},t)\Big) \\ & + \Big(-\frac{\gamma(t)^4}{24} + \frac{\gamma(t)^3}{3} - \frac{11\gamma(t)^2}{12} + \gamma(t)\Big)u(z_{3l},t) + \Big(\frac{\gamma(t)^4}{8} + \frac{3\gamma(t)^2}{2} - \frac{5\gamma(t)^3}{2}\Big)u(z_{3l+1},t) \\ & + \Big(-\frac{\gamma(t)^4}{8} + \frac{2\gamma(t)^3}{3} - \frac{3\gamma(t)^2}{4}\Big)u(z_{3l+2},t) + \Big(\frac{\gamma(t)^4}{24} - \frac{\gamma(t)^3}{6} + \frac{\gamma(t)^2}{6}\Big)u(z_{3l+3},t), \end{aligned} \tag{2.5}$$

3. Implementation

The domain $[0,1]\times[0,T]$ divided into an $M \times N$ mesh with spatial step size $\frac{1}{h}$ in x direction and time step size $k = \frac{T}{N}$, for $i = 1..M-1$ by multiplying Eq. (5) by 1 , 10 , and 1 in points x_{i-1}, x_i, and x_{i+1}, respectively. Then we have:

$$u_t|_{i-1} + 10u_t|_i + u_t|_{i+1} = u_{xx}|_{i-1} + 10u_{xx}|_i + u_{xx}|_{i+1} + r(t)(\phi(x,t)|_{i-1} + 10\phi(x,t)|_i + \phi(x,t)|_{i-1}), \tag{3.1}$$

substituting (11) in (16) gives:

$$\begin{aligned} \frac{h^2}{12\Delta t}(u_{i-1}^{n+1} + 10u_i^{n+1} + u_{i+1}^{n+1}) = {} & ((1 + \frac{h^2}{12\Delta t})u_{i+1}^n + (\frac{5h^2}{6\Delta t} - 2)u_i^n + (1 + \frac{h^2}{12\Delta t})u_{i-1}^n) \\ & + r^n\frac{h^2}{12}\big(\phi_{i-1}^n + 10\phi_i^n + \phi_{i-1}^n\big), \end{aligned} \tag{3.2}$$

on the other hand, for $i = 0$ or $i = M$ from boundary conditions we have:

$$\frac{g_0^{n+1}}{E^{n+1}}v^{n+1} - u_0^{n+1} = 0, \frac{g_1^{n+1}}{E^{n+1}}v^{n+1} - u_M^{n+1} = 0, \tag{3.3}$$

where $v^n = \int_0^{s(t_n)} u(x,t_n)dx$ can be obtained from (15) so we get system of equations bellow

$$A^n U^{n+1} = BU^n + r^n C\Phi^n,$$

where: $U^n = [u_0^n, u_1^n, ..., u_M^n]^T$, $\Phi^n = [\phi_0^n, \phi_1^n, ..., \phi_M^n]^T$, $A^n = A' + R^n$,

$$A' = \begin{bmatrix} -1 & 0 & 0 & ... & 0 & 0 & 0 \\ \frac{h^2}{12k} & \frac{5h^2}{6k} & \frac{h^2}{12k} & 0 & 0 & 0 & ... \\ 0 & \frac{h^2}{12k} & \frac{5h^2}{6k} & \frac{h^2}{12k} & 0 & 0 & ... \\ & \cdots & ... & ... & & & \\ ... & 0 & 0 & 0 & \frac{h^2}{12k} & \frac{5h^2}{6k} & \frac{h^2}{12k} \\ 0 & 0 & 0 & ... & 0 & 0 & -1 \end{bmatrix}, C = \frac{h^2}{12} \begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 1 & 10 & 1 & 0 & 0 & 0 & ... \\ 0 & 1 & 10 & 1 & 0 & 0 & ... \\ & \cdots & & ... & & & \\ & & & & \cdots & & \\ ... & 0 & 0 & \ddot{0} & 1 & 10 & 1 \\ 0 & 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 1+\frac{h^2}{12k} & \frac{5h^2}{6k}-2 & 1+\frac{h^2}{12k} & 0 & 0 & 0 & ... \\ 0 & 1+\frac{h^2}{12k} & \frac{5h^2}{6k}-2 & 1+\frac{h^2}{12k} & 0 & 0 & ... \\ & \cdots & ... & & ... & & \\ ... & 0 & 0 & 0 & 1+\frac{h^2}{12k} & \frac{5h^2}{6k}-2 & 1+\frac{h^2}{12k} \\ 0 & 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix},$$

$$R^n(i,j) = \frac{1}{E^{n+1}} \begin{cases} \frac{3hg_0^{n+1}}{4} & i = 1, j = 3k < 3l \\ \frac{9hg_0^{n+1}}{8} & i = 1, (j = 3k+1 < 3l \text{ or } j = 3k+2 < 3l) \\ \frac{3hg_1^{n+1}}{4} & i = M+1, j = 3k < 3l \\ \frac{9hg_1^{n+1}}{8} & i = M+1, (j = 3k+1 < 3l \text{ or } j = 3k+2 < 3l) \\ g_0^{n+1}\left(-\frac{\gamma(t)^4}{24} + \frac{\gamma(t)^3}{3} - \frac{11\gamma(t)^2}{12} + \gamma(t) + \frac{3h}{8} \right) & i = 1, j = 3l \\ g_0^{n+1}\left(\frac{\gamma(t)^4}{8} + \frac{3\gamma(t)^2}{2} - \frac{5\gamma(t)^3}{2} \right) & i = 1, j = 3l+1 \\ g_0^{n+1}\left(-\frac{\gamma(t)^4}{8} + \frac{2\gamma(t)^3}{3} - \frac{3\gamma(t)^2}{4} \right) & i = 1, j = 3l+2 \\ g_0^{n+1}\left(\frac{\gamma(t)^4}{24} - \frac{\gamma(t)^3}{6} + \frac{\gamma(t)^2}{6} \right) & i = 1, j = 3l+3 \\ g_1^{n+1}\left(-\frac{\gamma(t)^4}{24} + \frac{\gamma(t)^3}{3} - \frac{11\gamma(t)^2}{12} + \gamma(t) + \frac{3h}{8} \right) & i = M+1, j = 3l \\ g_1^{n+1}\left(\frac{\gamma(t)^4}{8} + \frac{3\gamma(t)^2}{2} - \frac{5\gamma(t)^3}{2} \right) & i = M+1, j = 3l+1 \\ g_1^{n+1}\left(-\frac{\gamma(t)^4}{8} + \frac{2\gamma(t)^3}{3} - \frac{3\gamma(t)^2}{4} \right) & i = M+1, j = 3l+2 \\ g_1^{n+1}\left(\frac{\gamma(t)^4}{24} - \frac{\gamma(t)^3}{6} + \frac{\gamma(t)^2}{6} \right) & i = M+1, j = 3l+3 \\ 0 & i \neq 1, i \neq M+1 \end{cases}.$$

which can be solved using multigrid algorithm that will be explained in next section.

In order to obtaining $p(t)$ we use relation $p(t) = -\frac{r'(t)}{r(t)}$, so for $r'(t)$ we have

$$(r')^1 = \frac{-25r^1 + 48r^2 - 36r^3 + 16r^4 - 3r^5}{12k}, \tag{3.4}$$

for $n = 2, \ldots, N-2$

$$(r')^n = \frac{r^{n-2} - 8r^{n-1} + 8r^{n+1} - r^{n+2}}{12k}, \tag{3.5}$$

$$(r')^{N-1} = \frac{3r^{N-5} - 16r^{N-4} + 36r^{N-3} - 48r^{N-2} + 25r^{N-1}}{12k}, \tag{3.6}$$

$$(r')^N = \frac{3r^{N-4} - 16r^{N-3} + 36r^{N-2} - 48r^{N-1} + 25r^N}{12k}. \tag{3.7}$$

4. Multigrid algorithm

In this part we should reviwe the general algorithm of multigrid[1]. We introduce Ω_h as the original and the finest uniform grid with mesh size h. the general multigrid cycle algorithm is defined bellow

Multigrid cycle $u_k^{m+1} = MGCYC(k, \gamma, u_k^m, L_k, f_k, v_1, v_2)$
(1) Presmoothing
- Compute $\bar{u}_k^m$ by applying $v_1 \geq 0$ smoothing steps to u_k^m

$$\bar{u}_k^m = SMOOTH^{v_1}(u_k^m, L_k, f_k)$$

(2) Coarse grid correction
- Compute the defect $\bar{d}_k^m = f_k - L_k \bar{u}_k^m$
- Restrict the defect $\bar{d}_{k-1}^m = I_k^{k-1} \bar{d}_k^m$
- Compute an approximate solution $\hat{v}_{k-1}^m$ of the defect equation on Ω_{k-1}

$$L_{k-1} \hat{v}_{k-1}^m = \bar{d}_{k-1}^m \tag{4.1}$$

by

If k = 1, use a direct or fast iterative solver for (23).
If $k > 1$, solve (23) approximately by performing $\gamma \geq 1$ k-grid cycles using the zero grid function as a first approximation

$$\hat{v}_{k-1}^m = MGCYC(k-1, \gamma, 0, L_{k-1}, \bar{d}_{k-1}^m, v_1, v_2)$$

-Interpolate the correction $\hat{v}_k^m = I_{k-1}^k \hat{v}_{k-1}^m$
-Compute the corrected approximation on Ω_k $u_k^{m,afterCGC} = \bar{u}_k^m + \hat{v}_k^m$
(3)Postsmoothing
- Compute u_k^{m+1} by applying $v_2 \geq 0$ smoothing steps to $u_k^{m,afterCGC}$

$$u_k^{m+1} = SMOOTH^{v_2}(u_k^{m,afterCGC}, L_k, f_k)$$

We refer to the case $\gamma = 1$ as V-cycles and to $\gamma = 2$ as W-cycles. The number γ is also called cycle index. We choose Gauss Seidel as smoother and restriction operator I_h^{2h} where the interpolation operator is chosen to be $I_{2h}^h = 2(I_h^{2h})^T$ where,

$$I_h^{2h} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & \dots & \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \dots & \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ \dots & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}.$$

Table 1: A comparison between absolute error obtained by the methods of [3-4] and the presented method when $T = 0.5$, $h = \frac{1}{50}$, $k = 0.0001$ for $w(x, T)$.

x	Exact w	Method of [3]	Method of [4]	**Presented method by v-cycle** $v_1 = v_2 = 4$
0.10	1.732899235	3.1×10^{-3}	6.5×10^{-4}	1.11675×10^{-8}
0.20	1.663587781	3.1×10^{-3}	6.7×10^{-4}	2.5928×10^{-8}
0.30	1.463710429	2.8×10^{-3}	6.5×10^{-4}	4.15869×10^{-8}
0.40	1.168971399	2.8×10^{-3}	6.6×10^{-4}	5.55161×10^{-8}
0.50	0.8243606352	2.6×10^{-3}	6.7×10^{-4}	6.54559×10^{-8}
0.60	0.4797498712	2.4×10^{-3}	6.3×10^{-4}	6.95088×10^{-8}
0.70	0.1850108408	2.6×10^{-3}	6.1×10^{-4}	6.6113×10^{-8}
0.80	-0.01486651063	2.7×10^{-3}	6.2×10^{-4}	5.40037×10^{-8}
0.90	-0.08417796478	2.9×10^{-3}	6.4×10^{-4}	3.21726×10^{-8}
CPU Time		14.9 s	15.8 s	2.6694 s

Table 2: A comparison between absolute error obtained by the methods of [3-4] and the presented method when $T = 0.5$, $h = \frac{1}{50}$, $k = 0.0001$ for $p(t)$.

t	Exact p	Method of [3]	Method of [4]	Presented method by v-cycle $v_1 = v_2 = 4$
0.05	0.900000	5.4×10^{-3}	8.7×10^{-4}	9.91214×10^{-5}
0.10	0.800000	5.3×10^{-3}	8.6×10^{-4}	1.00671×10^{-4}
0.15	0.700000	5.3×10^{-3}	8.8×10^{-4}	1.02775×10^{-4}
0.20	0.600000	5.2×10^{-3}	8.8×10^{-4}	1.07134×10^{-4}
0.25	0.500000	5.1×10^{-3}	8.6×10^{-4}	1.10997×10^{-4}
0.30	0.400000	5.0×10^{-3}	8.8×10^{-4}	1.17341×10^{-4}
0.35	0.300000	5.0×10^{-3}	8.7×10^{-4}	1.23845×10^{-4}
0.40	0.200000	4.9×10^{-3}	8.6×10^{-4}	1.30815×10^{-4}
0.45	0.100000	4.8×10^{-3}	8.5×10^{-4}	1.40057×10^{-4}
0.50	0.000000	4.7×10^{-3}	8.4×10^{-4}	1.49688×10^{-4}

5. Numerical results

We tested the accuracy and stability of the method presented in this paper by performing the mentioned method for fixed values of it's parameter. To show the efficiency and accuracy of presented scheme on the one-dimensional inverse problem, one example is presented. For that example, the exact solution is available. Consider (1)-(4) with: $\phi(x,t) = (\pi^2+2t)\exp(t)\cos(\pi x)+2\exp(t)xt$, and the boundary conditions: $g_0(t) = \exp(t)$ and $g_1(t) = 0$, with the initial condition: $f(x) = x + \cos(\pi x)$ and at last : $s(t) = 0.5(1 + \sqrt{t})$ and

$$E(t) = \exp(t)\left(\frac{\sin(0.5\pi(1+\sqrt{t}))}{\pi} + \frac{(1+\sqrt{t})^2}{8}\right) .$$

For which the exact solution is $w(x,t) = \exp(t)(\cos(\pi x) + x)$ and $p(t) = 1 - 2t$. All program implemented in c++ by using Armadillo package[2]. By implementing the numerical techniques, we produce the outcomes given Tables 1 and 2.

6. Conclusion

In this paper, we proposed an accurate and fast method based on compact finite difference scheme and multigrid algorithm for solving the one-dimensional inverse problem concerning diffusion equation with source control parameter. The accuracy and stability of the method was presented in one example and some other methods was compared with it.

References

[1] U. Trottenberg, C. W. Oosterlee, A. Schuller, *Multigrid*, AcademicPress, NewYork, 2001.
[2] C. Sanderson, R. Curtin, Armadillo: a template-based C++ library for linear algebra, *JOSS*, 1 (2016), 26–26.
[3] M. Dehghan, An inverse problem of finding a source parameter in a semilinear parabolic equation, *APPL MATH MODEL*, 25 (2001), 743–754.
[4] M. Dehghan, Numerical techniques for a parabolic equation subject to an overspecified boundary condition, *APPL MATH COMPUT*, 132 (2002), 299–313.