Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(4): 2136–2156.
DOI: 10.3934/era.2026096
Received: 20 December 2025
Revised: 24 February 2026
Accepted: 03 March 2026
Published: 10 March 2026

Research article

A computational approach to an inverse source problem for a kinetic equation with gradient-type boundary data and an interior point observation

İsmet Gölgeleyen[1], **Muhammed Hasdemir**[2,*] **and Özlem Kaytmaz**[1]

[1] Department of Mathematics, Faculty of Science, Zonguldak Bülent Ecevit University, Zonguldak 67100, Türkiye

[2] Aydın Vocational School of Health Services, Aydın Adnan Menderes University, Aydın 09020, Türkiye

* **Correspondence:** Email: muhammed.hasdemir@adu.edu.tr.

Abstract: In this work, we consider an inverse problem for a stationary kinetic equation. Our aim is to determine the source function from boundary measurements together with additional information provided at an interior point of the domain. Unlike existing works, the boundary information comprises not the direct problem solution itself, but the gradients the gradients of the solution and the source function are prescribed on the boundary. We develop a numerical algorithm based on a hybrid strategy that combines the finite-difference method with a bilinear interpolation polynomial approximation. The performance of the proposed approach is demonstrated through several numerical experiments, and the results are reported comparatively via graphs and tables. The numerical tests indicate that the reconstruction errors for the unknown functions remain sufficiently small.

Keywords: kinetic equation; inverse problem; hybrid algorithm; interior point observation; Tikhonov regularization

1. Introduction

Kinetic equations provide a statistical description of many-body systems in terms of the particle distribution function in the phase space. Originating from the classical ideas of Maxwell and Boltzmann and later shaped by developments in plasma and kinetic theory, this framework has become a standard modeling approach in rarefied gas dynamics and reacting gases, as well as in plasma physics and astrophysical settings such as stellar and galactic dynamics [1–4]. Kinetic models are used in a wide range of scientific and technological applications, including semiconductor device modeling, turbulence and fluid mechanics, stochastic dynamical systems, traffic flow, and biological processes such as chemotaxis and immunology [5–7]. In this context, inverse problems for kinetic equations focus on identifying physically meaningful quantities, such as interaction potentials, scattering properties, or

internal source terms, from indirect measurements [8–10].

Inverse problems for kinetic equations were theoretically studied by Amirov [9] and Anikonov [10]. Owing to the overdetermined nature of these problems, they were first reduced to higher-order boundary value problems with Dirichlet-type data on the distribution function. In these problems, the trace of the solution on the entire boundary plays the role of overdetermined Dirichlet data, while the unknown source is extended to depend on both position and velocity and is required to satisfy a mixed second-order differential constraint. With this technique, the overdetermined inverse problem is transformed into a determined problem, which yields uniqueness, and in several cases, existence results for various kinetic-type equations, such as Liouville, Boltzmann, Vlasov, and quantum-kinetic equations [9].

The well-posedness of these types of inverse problems for kinetic and transport equations was investigated in [11–16] where numerical solutions were obtained using symbolic computation based on Galerkin approximations and employing various numerical approaches, such as finite-difference and polynomial approximations. Gölgeleyen and Hasdemir [17] proposed a numerical solution algorithm for an inverse source problem associated with a non-stationary kinetic equation. Their approach is based on a hybrid scheme that combines an implicit finite-difference method with Lagrange polynomial interpolation to reconstruct both the unknown solution and source term from the initial, boundary, and final-time data. They performed a Von Neumann stability analysis and numerically showed that the algorithm yields accurate reconstructions with small relative errors and is robust to noisy boundary data. In recent years, efficient numerical algorithms for inverse problems associated with various differential equations have been proposed in [18, 19]. Although it is observed in the literature that the Galerkin method provides highly precise results, it requires the problems to possess homogeneous Dirichlet boundary conditions, which acts as a restrictive factor in practical applications. Since the attainable accuracy depends sensitively on the equation type and the formulation of the problem (e.g., coefficients, boundary data and the observation setting), numerical results reported under different model configurations and error metrics are not always directly comparable.

Using a similar methodology, the solvability and approximate solution of a general transport equation were studied in [20], where a detailed numerical analysis was presented. As for the uniqueness and stability results for the transport equations we refer to [21–25] where the key tool is a Carleman estimate. In [26], a globally convergent so-called convexification numerical method was developed for a coefficient inverse problem for the radiative transport equation, and its convergence analysis was provided.

Here, we deal with an inverse source problem for a kinetic equation by prescribing the gradient of the unknown function on the boundary, the gradient of the source function with respect to the velocity variables on the boundary, and one interior value of the solution, rather than prescribing the solution itself, as in [11–16]. Moreover, a uniqueness theorem is obtained for this problem without imposing any geometric restrictions on the domain, provided that the full boundary gradient of the source is available. In previous studies, the geometry of the domain was given in the form of a Cartesian product.

2. Statement of the problem

In the present work, we consider the following kinetic equation:

$$Lu(x, v) \equiv \{H, u\} = \lambda(x, v) + F(x, v) \tag{2.1}$$

in the domain $\Omega = \{(x, v) : x \in D \subset \mathbb{R}^n, v \in G \subset \mathbb{R}^n\}$ with the boundary conditions

$$\nabla u|_{\partial\Omega} = u_0, \qquad \nabla_v \lambda|_{\partial\Omega} = \lambda_0, \tag{2.2}$$

and the interior data

$$u(x_0, v_0) = u_1. \tag{2.3}$$

Here, the Poisson bracket is defined by

$$\{H, u\} = \sum_{i=1}^{n} \left(\frac{\partial H}{\partial v_i} \frac{\partial u}{\partial x_i} - \frac{\partial u}{\partial v_i} \frac{\partial H}{\partial x_i} \right).$$

The unknown $u(x, v)$ represents the stationary phase space density of particles at position x with velocity v. The function $H(x, v)$ is a given Hamiltonian, and the term $F(x, v)$ collects all known contributions to the right–hand side of the kinetic equation, such as prescribed external sources or modeled collision effects. The function $\lambda(x, v)$ is an unknown source term.

Problem 1. *Determine the pair of functions $u(x, v)$ and $\lambda(x, v)$ that satisfies Eq* (2.1) *with the boundary conditions* (2.2) *and the interior point data (2.3).*

The solvability of Problem 1 is established by the following theorem, see Amirov [9].

Theorem 1. *Assume that $H \in C^2\left(\overline{\Omega}\right)$ and the following inequalities hold for all $\xi \in \mathbb{R}^n$, $(x, v) \in \overline{\Omega}$:*

$$\sum_{i,j=1}^{n} \frac{\partial^2 H}{\partial v_i v_j} \xi^i \xi^j \geq \alpha \left|\xi\right|^2, \ \sum_{i,j=1}^{n} \frac{\partial^2 H}{\partial x_i x_j} \xi^i \xi^j \leq 0, \tag{2.4}$$

where α is a positive number. Moreover, let the function $\lambda(x, v)$ satisfy the following differential equation

$$\widehat{L}\lambda \equiv \sum_{i=1}^{n} \frac{\partial^2 \lambda}{\partial v_i \partial x_i} = 0. \tag{2.5}$$

Then, Problem 1 has at most one solution (u, λ) such that $u \in C^2\left(\overline{\Omega}\right)$, $\lambda \in C^2\left(\overline{\Omega}\right)$.

The proof of Theorem 1 is carried out in a manner similar to that in Amirov [9].

In classical boundary value problems for kinetic equations, uniqueness is often established through energy-type inequalities where one typically invokes a *Poincaré-Steklov inequality* to pass from a gradient-based quantity (e.g., $\|\nabla u\|$) to a bound involving the function itself (e.g., $\|u\|$). In the setting considered here, however, the availability of gradient/flux-type boundary data allows the boundary terms to be controlled directly, so this intermediate reduction is not essential. Consequently, the uniqueness arguments can be formulated more directly without relying on a separate step that converts gradient information into estimates for the trace or values of u.

From a physical viewpoint, problems with gradient-type boundary information constitute an important and practically relevant class. In many realistic measurement procedures, sensors do not record the field variable itself, but rather flux or normal-derivative quantities across the boundary (e.g., heat flux in conduction, current density in electromagnetism, or particle flux in transport). In some experimental settings, sensors measure boundary fluxes or normal derivatives rather than the field itself; hence, gradient-type boundary conditions can provide a closer match to physically accessible measurements and practical observation scenarios.

3. The numerical solution algorithm

In this section, to implement the numerical scheme explicitly, we consider the one-dimensional case ($n = 1$) and set $\Omega = (a, b) \times (c, d)$ and $a, b, c, d \in \mathbb{R}$. First, applying the operator $\widehat{L} = \frac{\partial^2}{\partial v \partial x}$ to both sides of (2.1) yields the auxiliary third-order partial differential equation:

$$H_{vvx}u_x + H_{vv}u_{xx} + H_v u_{xxv} - H_{xxv}u_v - H_{xx}u_{vv} - H_x u_{vxv} = F_{xv}. \tag{3.1}$$

Using the central finite-difference formulas in Eq (3.1), we obtain the following discrete version:

$$\begin{aligned}
&(H_{vvx})_{i,j}\frac{\left(\tilde{u}_{i+1,j} - \tilde{u}_{i-1,j}\right)}{2\Delta x} + (H_{vv})_{i,j}\frac{\left(\tilde{u}_{i+1,j} - 2\tilde{u}_{i,j} + \tilde{u}_{i-1,j}\right)}{(\Delta x)^2}\\
&+ (H_v)_{i,j}\frac{\left(2\tilde{u}_{i,j-1} - 2\tilde{u}_{i,j+1} + \tilde{u}_{i+1,j+1} + \tilde{u}_{i-1,j+1} - \tilde{u}_{i+1,j-1} - \tilde{u}_{i-1,j-1}\right)}{2(\Delta x)^2(\Delta v)}\\
&- (H_{xxv})_{i,j}\frac{\left(\tilde{u}_{i,j+1} - \tilde{u}_{i,j-1}\right)}{2\Delta v} - (H_{xx})_{i,j}\frac{\left(\tilde{u}_{i,j+1} - 2\tilde{u}_{i,j} + \tilde{u}_{i,j-1}\right)}{(\Delta v)^2}\\
&- (H_x)_{i,j}\frac{\left(2\tilde{u}_{i-1,j} - 2\tilde{u}_{i+1,j} + \tilde{u}_{i+1,j+1} - \tilde{u}_{i-1,j+1} + \tilde{u}_{i+1,j-1} - \tilde{u}_{i-1,j-1}\right)}{2(\Delta x)(\Delta v)^2} = (F_{xv})_{i,j}.
\end{aligned}$$

After the necessary simplifications, we write the following discrete equation:

$$\begin{aligned}
\widehat{L}Lu = {} & \tilde{u}_{i-1,j-1}(-k_1 + k_2) + \tilde{u}_{i-1,j}(-k_5 + k_3 - 2k_2) + \tilde{u}_{i-1,j+1}(k_1 + k_2) + \tilde{u}_{i,j-1}(2k_1 + k_6 - k_4)\\
& + \tilde{u}_{i,j}(-2k_3 + 2k_4) + \tilde{u}_{i,j+1}(-2k_1 - k_6 - k_4) + \tilde{u}_{i+1,j-1}(-k_1 - k_2) + \tilde{u}_{i+1,j}(k_5 + k_3 + 2k_2)\\
& + \tilde{u}_{i+1,j+1}(k_1 - k_2) = (F_{xv})_{i,j},\\
& i = \overline{1, I},\ j = \overline{1, J},
\end{aligned} \tag{3.2}$$

where the coefficients

$$\begin{aligned}
&k_1(i,j) = \frac{(H_v)_{i,j}}{2(\Delta x)^2(\Delta v)}, \qquad k_2(i,j) = \frac{(H_x)_{i,j}}{2(\Delta x)(\Delta v)^2}, \qquad k_3(i,j) = \frac{(H_{vv})_{i,j}}{(\Delta x)^2},\\
&k_4(i,j) = \frac{(H_{xx})_{i,j}}{(\Delta v)^2}, \qquad k_5(i,j) = \frac{(H_{vvx})_{i,j}}{2\Delta x}, \qquad k_6(i,j) = \frac{(H_{xxv})_{i,j}}{2\Delta v},
\end{aligned}$$

$I,\ J$ are positive integers and $\Delta x = \frac{(b-a)}{(I+1)}$, $\Delta v = \frac{(d-c)}{(J+1)}$ are step sizes in the directions $x,\ v$, respectively.

In order to incorporate the boundary conditions into the finite-difference scheme, we use the following approximations at the boundary data:

$$\begin{aligned}
&u_x(a,v) \approx \frac{-3\tilde{u}_{0,j} + 4\tilde{u}_{1,j} - \tilde{u}_{2,j}}{2\Delta x} = g_L\left(v_j\right), \qquad u_x(b,v) \approx \frac{3\tilde{u}_{I+1,j} - 4\tilde{u}_{I,j} + \tilde{u}_{I-1,j}}{2\Delta x} = g_R\left(v_j\right),\\
&u_v(x,c) \approx \frac{-3\tilde{u}_{i,0} + 4\tilde{u}_{i,1} - \tilde{u}_{i,2}}{2\Delta v} = h_B(x_i), \qquad u_v(x,d) \approx \frac{3\tilde{u}_{i,J+1} - 4\tilde{u}_{i,J} + \tilde{u}_{i,J-1}}{2\Delta v} = h_T(x_i),\\
&i = \overline{1, I+1},\ j = \overline{1, J+1}.
\end{aligned}$$

The approximate solution $\tilde{u}_{i,j}$ of Problem (1) is obtained at $I \times J$ mesh points of Ω by solving the matrix equation

$$\tilde{A}\widetilde{\mathbf{u}} + \mathbf{b} = \mathcal{F}, \tag{3.3}$$

where $\tilde{A}$ is a block tridiagonal banded matrix of the form

$$\tilde{A} = \begin{bmatrix} \mathcal{A}_1^{(1)} & \mathcal{B}_1^{(1)} & 0 & \cdots & 0 \\ C_1^{(2)} & \mathcal{A}_2^{(2)} & \mathcal{B}_2^{(2)} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & C_1^{(I-1)} & \mathcal{A}_2^{(I-1)} & \mathcal{B}_2^{(I-1)} \\ 0 & \dots & 0 & C_2^{(I)} & \mathcal{A}_3^{(I)} \end{bmatrix}_{IJ\times\, IJ}. \tag{3.4}$$

To maintain the narrative flow of the numerical algorithm, the explicit definitions of the submatrices $\mathcal{A}_k^{(i)}, \mathcal{B}_k^{(i)}, C_k^{(i)}$ and their corresponding algebraic coefficients are detailed in the Appendix.

In (3.3), the right-hand side $\mathcal{F}$ is a column vector given by

$$\mathcal{F} = [\mathcal{F}_{1,1}, \mathcal{F}_{1,2}, ..., \mathcal{F}_{1,J}, \mathcal{F}_{2,1}, \mathcal{F}_{2,2}, ..., \mathcal{F}_{2,J}, ..., \mathcal{F}_{I,1}, \mathcal{F}_{I,2}, ..., \mathcal{F}_{I,J}]^T,$$

where $\mathcal{F}_{i,j} = F_{xv}(x_i, v_j)$ for $i = \overline{1, I}$ and $j = \overline{1, J}$. The solution vector $\widetilde{\mathbf{u}}$ of size $IJ \times 1$ is defined as

$$\widetilde{\mathbf{u}} = [\tilde{u}_{1,1}, \tilde{u}_{1,2}, ..., \tilde{u}_{1,J}, \tilde{u}_{2,1}, \tilde{u}_{2,2}, ..., \tilde{u}_{2,J}, ..., \tilde{u}_{I,1}, \tilde{u}_{I,2}, ..., \tilde{u}_{I,J}]^T.$$

To ensure a unique solution, we incorporated the interior point condition $u(x_0, v_0) = u_1$ directly into our linear system. As shown in Figure 1, this forces the numerical solution to pass exactly through the prescribed value, which stabilizes the entire system:

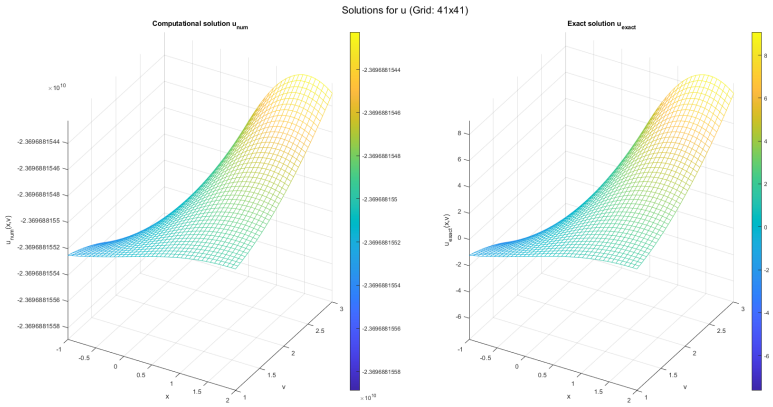


Figure 1. Effect of the interior point condition $u(x_0, v_0) = u_1$ on the numerical solution.

Furthermore, a small spatial discretization error naturally remains even in the noise-free case. Additionally, if the interior point (x_0, v_0) is slightly shifted within the grid, the solution exhibits a mild change, reflecting its sensitivity to positional uncertainty. To systematically evaluate robustness against such perturbations, we present Monte Carlo simulations with additive Gaussian noise in the numerical examples section.

Now that the numerical solution u is known, next we will determine the function $\lambda(x, v)$ from the equation

$$\lambda(x, v) \approx \{H, u\} - F(x, v)$$

with the theoretical conditions

$$\frac{\partial^2 \lambda}{\partial v \partial x} = 0, \left.\frac{\partial \lambda}{\partial v}\right|_{\partial\Omega} = \lambda_0.$$

We compute $\lambda(x, v)$ by the following five steps.

Step 1. For each 2×2 mesh, we take the four corner values of the numerical solution $\widetilde{\mathbf{u}} : (Q_{11}, Q_{12}, Q_{21}, Q_{22})$. We assume that, inside this cell, $u(x, v)$ can be approximated by a bilinear surface

$$P(x, v) = a_0 + a_1 x + a_2 v + a_3 xv,$$

which interpolates these four corner values. This $P(x, v)$ is used as a local model of $u(x, v)$ in that small cell, see, e.g., [27, 28].

Step 2. We compute its analytical derivatives:

$$\frac{\partial P(x, v)}{\partial x} = a_1 + a_3 v, \ \frac{\partial P(x, v)}{\partial v} = a_2 + a_3 x.$$

These formulas provide smooth approximations of u_x and u_v inside each cell. Using them in the Poisson bracket $\{H, u\} = H_v u_x - H_x u_v$, we then construct the source

$$\lambda(x, v) \approx \{H, u\} - F(x, v).$$

Step 3. Motivated by the condition $\lambda_{xv} = 0$ in Ω, we assume that λ is of the separated form:

$$\lambda(x, v) = A(x) + B(v).$$

Thus, the problem reduces to determining the functions $A(x)$ and $B(v)$. In the discrete setting, the unknowns are the grid values of these functions, denoted by the vectors $A_i \approx A(x_i)$ and $B_j \approx B(v_j)$ so that $\lambda_{ij} \approx A_i + B_j$ at each grid point (x_i, v_j).

Step 4. Combine the smoothed data from Steps 1–3, the theoretical model $\lambda(x, v) = A(x) + B(v)$, and all additional conditions into a single large linear system

$$Mz \approx y,$$

where $z = [A_1, \ldots, A_I, B_1, \ldots, B_J]^T$.

We assemble the matrix M and the vector y from the following sets of equations.

First, for each cell center (i, j), we impose

$$\frac{A_i + A_{i+1}}{2} + \frac{B_j + B_{j+1}}{2} \approx \lambda_{ij},$$

where λ_{ij} denotes the value of λ at the center of the cell associated with (x_i, v_j). This large collection of equations ensures that the reconstructed λ is consistent with the information coming from the numerical solution u.

On the v-boundaries, we add the discrete equations

$$\frac{B_{j+1} - B_j}{\Delta v} \approx \lambda_0, \quad \frac{B_j - B_{j-1}}{\Delta v} \approx \lambda_0, \quad \frac{B_{j+1} - B_{j-1}}{2\Delta v} \approx \lambda_0,$$

which represent forward, backward, and central finite–difference approximations of the boundary condition

$$\partial_v \lambda|_{\partial\Omega} = \lambda_0.$$

To stabilize the reconstruction, we add a Tikhonov regularization term that penalizes roughness in the recovered λ. In discrete form, this is implemented by the second-order difference relations

$$A_{i+1} - 2A_i + A_{i-1} \approx 0, \qquad B_{j+1} - 2B_j + B_{j-1} \approx 0$$

for interior indices $2 \le i \le I - 1$ and $2 \le j \le J - 1$. This corresponds to a standard smoothness-promoting Tikhonov regularization for discrete inverse problems [29, Eq (2.2.2)]. The regularization parameter α is chosen by the L-curve criterion [30]. We test a logarithmically-spaced set of candidate values $\{\alpha_k\}$ over the range $[10^{-3}, 10^2]$, and for each α_k we solve the corresponding regularized least-squares system. We then compute the residual norm (measuring data fit) and the solution norm (measuring smoothness via the discrete second-difference operator). The optimal parameter α_{opt} is selected by maximizing the discrete curvature at the corner of the L-curve in logarithmic coordinates, and α_{opt} is used in the final reconstruction.

The effect of the smoothing and Tikhonov regularization is illustrated in Figure 2; the unregularized reconstruction exhibits strong spurious oscillations, whereas the regularized one is much smoother while preserving the main trend of $\lambda(x, v)$.

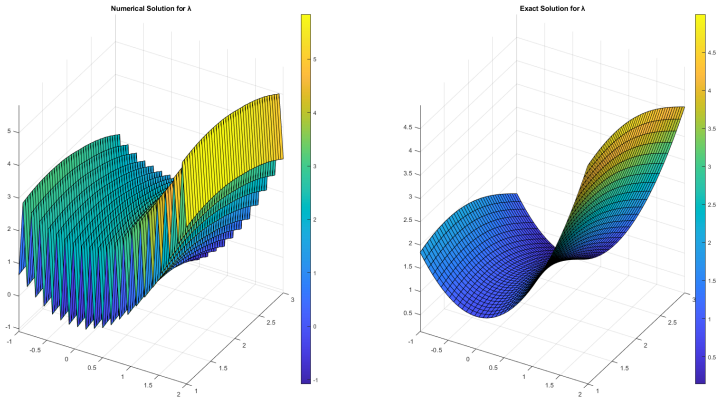


Figure 2. Effect of smoothing and Tikhonov regularization on the reconstruction of $\lambda(x, v)$.

Step 5. Finally, we solve the augmented linear system $Mz = y$, which incorporates all data, boundary, and Tikhonov regularization constraints. From the resulting coefficient vector z we extract the discrete profiles A_i and B_j and reconstruct the final, regularized solution for $\lambda(x, v)$ on the grid via

$$\lambda(x_i, v_j) = A_i + B_j.$$

Algorithm 1 The numerical solution algorithm

Input: Ω, $\Delta x, \Delta v$, $H(x, v)$, boundary data g_L, g_R, h_B, h_T, $F(x, v)$ (or F_{xv}), interior point (x_0, v_0, u_1), and λ_0.
Initialize: Grid points (x_i, v_j) for $i = 1..I$ and $j = 1..J$.
procedure MATRIX ASSEMBLY
    for each interior node (i, j) **do**
        Compute $k_1(i, j), \ldots, k_6(i, j)$ from H.
        Update the corresponding entries of $\tilde{A}$ using (3.2).
        Set $\mathcal{F}_{i,j} = F_{xv}(x_i, v_j)$.
    end for
    Incorporate boundary conditions into $\tilde{A}$ and $\mathbf{b}$ (one-sided differences).
    Enforce $u(x_0, v_0) = u_1$ by replacing the corresponding row.
end procedure
Solve: $\tilde{A}\widetilde{\mathbf{u}} = \mathcal{F} - \mathbf{b}$.
procedure RECONSTRUCT $\lambda(x, v)$
    for each 2×2 cell **do**
        Build bilinear $P(x, v)$ from $\tilde{u}$ and compute u_x, u_v.
        Set $\lambda^{\text{raw}}(x, v) \approx \{H, u\} - F, \quad \{H, u\} = H_v u_x - H_x u_v$.
    end for
    Assume $\lambda(x, v) = A(x) + B(v)$.
    Assemble $Mz \approx y$ (cell-center data, $\partial_v \lambda|_{\partial\Omega} = \lambda_0$ and Tikhonov smoothness).
    Choose α via L-curve and solve for $z = [A_1..A_I, B_1..B_J]^T$.
end procedure
Output: $\tilde{u}(x_i, v_j)$ and $\lambda(x_i, v_j)$.

4. Numerical experiment

In this section, we present two numerical examples to illustrate the performance of the proposed algorithm. For each example, we compare the numerical and exact solutions for u and λ, and show the corresponding error measures. We report the maximum absolute error (MaxAbsErr) and the relative errors (RelErr) in the discrete L^2 and L^∞ norms on an $(I \times J)$ uniform grid with spacings Δx and Δv:

$$\text{for } q \in \{u, \lambda\}, \qquad e_{ij}^{(q)} = q_{ij}^{\text{num}} - q_{ij}^{\text{ex}}, \qquad 1 \le i \le I,\ 1 \le j \le J,$$

$$\text{MaxAbsErr}(q) = \|e^{(q)}\|_\infty := \max_{i,j} \left|e_{ij}^{(q)}\right|,$$

$$\|w\|_2 := \left(\sum_{i=1}^{I} \sum_{j=1}^{J} |w_{ij}|^2 \, \Delta x \, \Delta v \right)^{1/2},$$

$$\text{RelErr}_2(q) = \frac{\|e^{(q)}\|_2}{\|q^{\text{ex}}\|_2}, \qquad \text{RelErr}_\infty(q) = \frac{\|e^{(q)}\|_\infty}{\|q^{\text{ex}}\|_\infty},$$

where $\|\cdot\|_2$ is the standard grid-weighted discrete analogue of the continuous L^2 norm used in finite-difference error analysis, and the ratios above correspond to the usual normwise relative error measures [31, 32].

Example 1. *Let us consider the problem of finding the pair* (u, λ) *in* $\Omega = (1, 3) \times (1, 3)$ *for Eq* (2.1)*, where*

$$H(x, v) = \cos(x^2) - \frac{v^2}{2}$$

and

$$F(x, v) = \frac{x^2}{2} - \cos(v^3) - 4xv^3 - 2xv + 8x^3 v \sin(x^2),$$

the boundary conditions are

$$u_x(1, v) = 4v^2 + 2, \qquad u_x(3, v) = 12v^2 + 6, \qquad u_v(x, 1) = 4x^2, \qquad u_v(x, 3) = 12x^2,$$
$$\lambda_v(x, 1) = -1, \qquad \lambda_v(x, 3) = -3,$$

and the interior condition $u(2, 2) = 36$*.*

The exact solution of the problem is

$$u\,(x, v) = (x^2)(v^2 + 1) + (x^2v^2), \qquad \lambda\,(x, v) = \cos(v^3) - \frac{x^2}{2}.$$

Tables 1 and 2 summarize the noise-free results for different grid sizes. As the mesh is refined, the 1-norm condition number $\kappa_1(\tilde{A}) = \|\tilde{A}\|_1\|\tilde{A}^{-1}\|_1$ increases sharply (Table 1), while the error in u remains very small (below 10^{-7} in maximum absolute error). To assess the practical applicability and scalability of the proposed method, Table 2 reports the CPU times for various grid resolutions. When refining the grid from 41×41 to 401×401, the computational runtime increases from 12.25 s to 120.15 s. Over the same refinement, the total number of grid points increases by a factor of approximately 95.7, whereas the runtime increases by only about 9.8. This favorable scaling behavior indicates that the proposed method can handle high-resolution practical applications without incurring prohibitive computational costs.

Table 1. Noise-free performance for different grid sizes: error metrics for the solution u and estimated conditioning of the linear system.

Grid size		Solution u		$\kappa_1(\tilde{A})$
$(I \times J)$	$MaxAbsErr$	$RelErr_2$	$RelErr_\infty$	
41×41	1.97×10^{-10}	1.22×10^{-12}	1.15×10^{-12}	2.6×10^{10}
81×81	3.57×10^{-10}	1.72×10^{-12}	2.09×10^{-12}	1.23×10^{12}
121×121	1.05×10^{-9}	5.00×10^{-12}	6.19×10^{-12}	2.05×10^{13}
401×401	9.53×10^{-8}	5.63×10^{-10}	5.57×10^{-10}	8.93×10^{16}

Table 2. Noise-free performance for different grid sizes: error metrics for the source λ and CPU time.

Grid size		Source λ		CPU time (s)
$(I \times J)$	$MaxAbsErr$	$RelErr_2$	$RelErr_\infty$	
41×41	2.16×10^{-1}	2.62×10^{-2}	3.92×10^{-2}	12.25
81×81	8.92×10^{-2}	7.36×10^{-3}	1.62×10^{-2}	21.18
121×121	5.65×10^{-2}	2.88×10^{-3}	1.08×10^{-2}	38.17
401×401	2.27×10^{-3}	2.57×10^{-4}	4.13×10^{-4}	120.15

The sparsity pattern of the coefficient matrix A used in the numerical solution of u is shown in Figure 3. Moreover, in Figures 4–11 the numerical solutions for u and λ are compared with the corresponding exact solutions through surface plots and cross-sections. In Figure 12, the spatial distributions of the absolute errors for the numerical solution u and the reconstructed source λ over the computational domain are displayed.

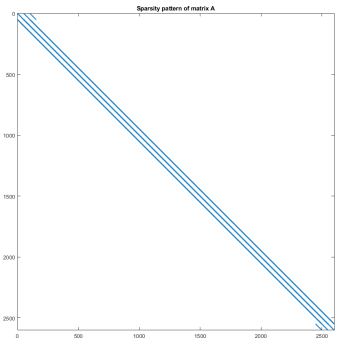


Figure 3. Sparsity pattern of the coefficient matrix A.

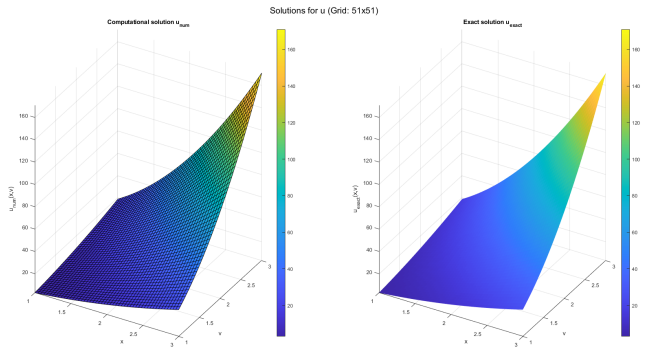


Figure 4. Numerical solution $u_{\text{num}}(x, v)$ on the computational grid.

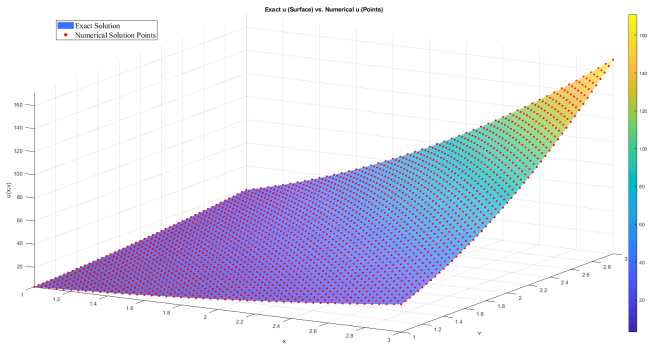
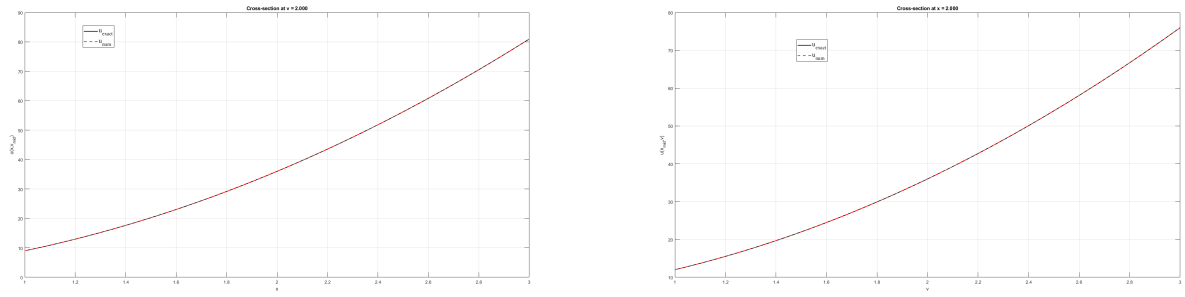


Figure 5. Comparison of the exact solution $u_{\text{exact}}(x, v)$ (surface) and the numerical values (points).



(a) (b)

Figure 6. Cross–sections of $u(x, v)$ and comparison of exact and numerical solutions. (a) Cross–section for a fixed value of v. (b) Cross–section for a fixed value of x.

Figure 7 illustrates the influence of interior-point noise on the numerical solution u. The error measures increase monotonically with the noise level and show an approximately linear trend on the log-log scale.

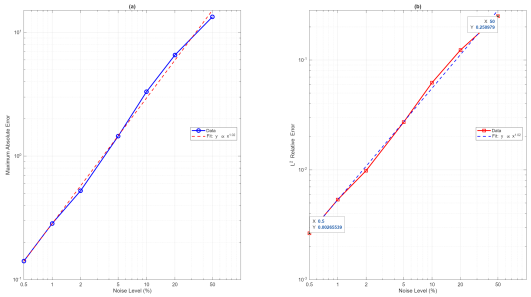


Figure 7. Interior-point noise sensitivity on a 81×81 grid (Monte Carlo). (a) Maximum absolute error versus noise level. (b) Relative L^2 error versus noise level. Dashed lines show power-law fits obtained from a log-log regression.

For the numerical example, the regularization parameter is selected by the L-curve criterion; as shown in Figure 8, the corner identified via maximum discrete curvature yields $\alpha_{opt} = 7.0486 \times 10^{-2}$, which is used in the final reconstruction.

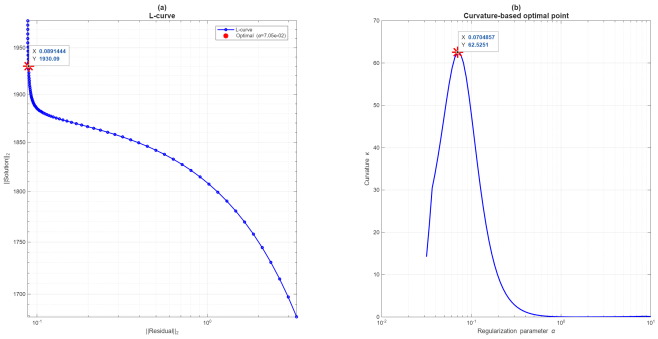


Figure 8. (a) L-curve illustrating the trade-off between the residual norm and the solution (smoothness) norm. (b) Selection of the regularization parameter via the L-curve criterion; the optimal value is chosen at the corner.

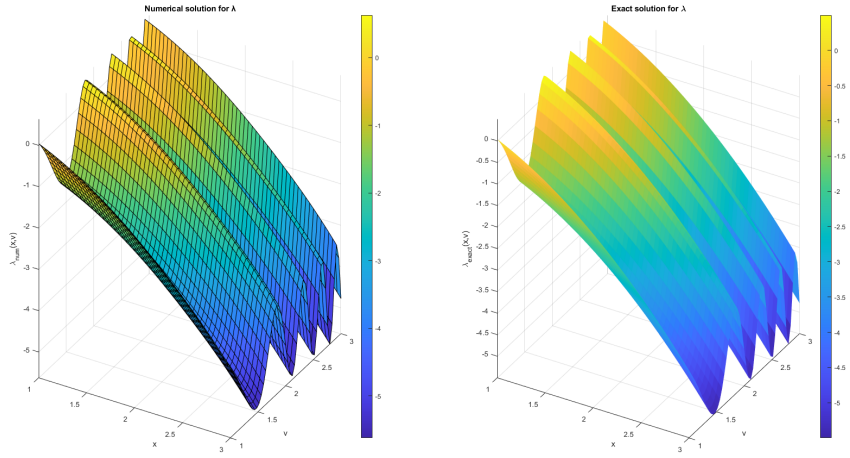
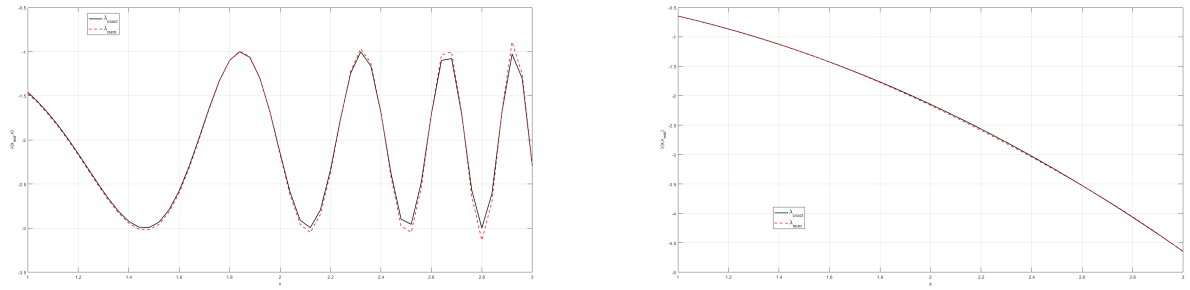


Figure 9. Numerical and exact solutions for $\lambda(x, v)$ on the grid.

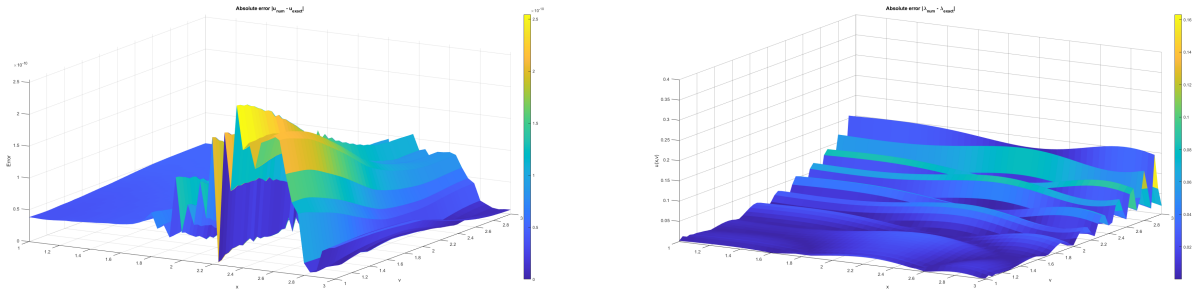


(a) (b)

Figure 10. Cross–sections of $\lambda(x, v)$ and comparison of exact and numerical solutions. (a) Cross–sections for fixed values of x. (b) Cross–sections for fixed values of v.

(a) (b)

Figure 11. (a) Correlation plot for u: numerical values versus exact values (points should lie close to the line $y = x$). (b) Correlation plot for $\lambda(x, v)$.



(a) (b)

Figure 12. Absolute error surfaces over the computational domain. (a) Absolute error of u. (b) Absolute error of λ.

Example 2. *Let us consider the problem of determining the functions u and λ in $\Omega = (-1, 2) \times (1, 3)$ for Eq* (2.1)*, where the Hamiltonian is*

$$H(x, v) = \frac{x^2}{2} + \cos v$$

and

$$F(x, v) = x(xe^{-v} - 2v \sin x) - \sin v - x^2 - \sin v(e^{-v} + v^2).$$

The boundary conditions are

$$u_x(-1, v) = g_L(v) = e^{-v} + v^2 \cos(1), \qquad u_x(2, v) = g_R(v) = e^{-v} + v^2 \cos(2),$$
$$u_v(x, 1) = h_B(x) = 2 \sin x - xe^{-1}, \qquad u_v(x, 3) = h_T(x) = 6 \sin x - xe^{-3},$$

while the derivative of λ at the v–boundaries is prescribed as

$$\lambda_v(x, 1) = \cos(1), \qquad \lambda_v(x, 3) = \cos(3)$$

and we are given the interior point data $u(0.5, 2) = 1.9854$.

In this example, the exact solution of the problem is given as follows:

$$u(x, v) = v^2 \sin x + e^{-v}x, \qquad \lambda(x, v) = x^2 + \sin v.$$

As shown in Table 3, the results for u are summarized in terms of the maximum absolute error, the relative L^2 and L^∞ errors, and $\kappa_1(\tilde{A})$. Table 4 reports the corresponding error metrics for λ together with the CPU time for each grid resolution.

Table 3. Noise-free performance for different grid sizes: error metrics for the solution u and estimated conditioning of the linear system.

Grid Size		Solution u		$\kappa_1(\tilde{A})$
$(I \times J)$	$MaxAbsErr$	$RelErr_2$	$RelErr_\infty$	
51×51	6.79×10^{-2}	5.63×10^{-3}	7.48×10^{-3}	2.60×10^{11}
81×81	2.83×10^{-2}	2.33×10^{-3}	3.11×10^{-3}	3.78×10^{12}
101×101	1.78×10^{-2}	1.49×10^{-2}	1.96×10^{-2}	1.70×10^{13}
121×121	1.21×10^{-2}	1.05×10^{-2}	1.33×10^{-2}	5.40×10^{13}
401×401	1.20×10^{-3}	9.88×10^{-5}	1.31×10^{-2}	8.20×10^{16}

Table 4. Noise-free performance for different grid sizes: error metrics for the source λ and CPU time.

Grid size		Source λ		CPU time (s)
$(I \times J)$	$MaxAbsErr$	$RelErr_2$	$RelErr_\infty$	
51×51	4.52×10^{-2}	9.15×10^{-3}	8.99×10^{-3}	18.95
81×81	2.00×10^{-2}	4.03×10^{-3}	3.99×10^{-3}	22.31
101×101	1.32×10^{-2}	2.64×10^{-3}	2.63×10^{-2}	30.53
121×121	9.15×10^{-3}	1.89×10^{-3}	1.83×10^{-3}	44.13
401×401	9.49×10^{-4}	1.86×10^{-4}	1.89×10^{-4}	230.25

Figures 13–19 present comparisons between the numerical and exact solutions for u and λ in Example 2, whereas Figure 20 displays the corresponding absolute error for both u and λ.

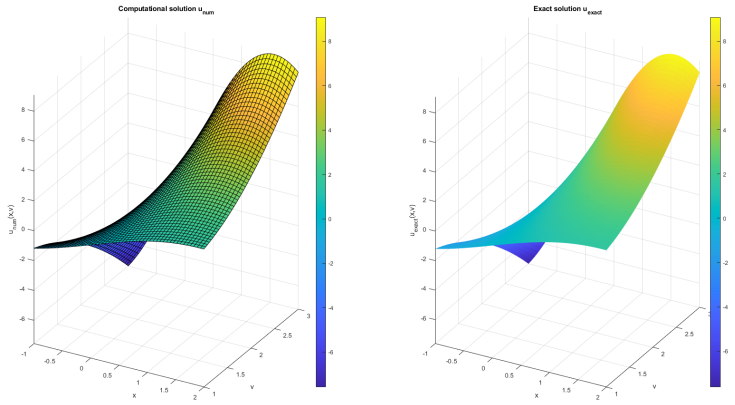


Figure 13. Comparison of the computational solution $u_{\text{num}}(x, v)$ (left) and the exact solution $u_{\text{exact}}(x, v)$ (right) over the computational domain.

(a) (b)

Figure 14. (a) Cross–section of $u(x, v)$ for a fixed value of v: comparison of the exact and numerical solutions. (b) Cross–section of $u(x, v)$ for a fixed value of x: comparison of the exact and numerical solutions.

To evaluate the robustness of the interior-point constraint, we performed a Monte Carlo noise sensitivity study on the 121×121 grid by perturbing the interior-point data with additive Gaussian noise at prescribed levels (0–50%) and repeating the experiment for 100 realizations. Figure 15 shows that

the reconstruction error increases with the noise level. The log–log slopes are approximately 0.81 for the maximum absolute error and 0.95 for the L^2 relative error, suggesting an almost linear increase in the L^2 relative error. At low noise levels, the solution remains stable (e.g., relative L^2 errors on the order of 10^{-3} at 0.5% noise), whereas larger perturbations progressively degrade accuracy.

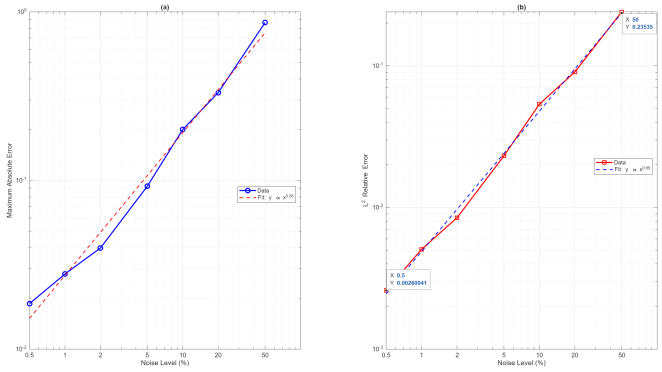


Figure 15. Interior-point noise sensitivity on a 121×121 grid (Monte Carlo). (a) Maximum absolute error versus noise level. (b) Relative L^2 error versus noise level. Dashed lines show power-law fits obtained from a log-log regression.

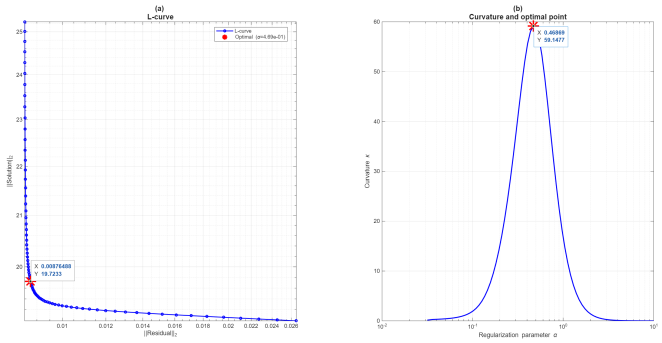


Figure 16. (a) L-curve illustrating the trade-off between the residual norm and the solution (smoothness) norm. (b) Selection of the regularization parameter via the L-curve criterion; the optimal value is chosen at the corner.

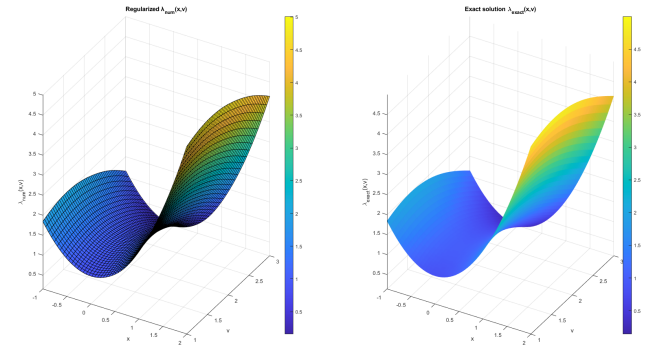


Figure 17. Comparison of the regularized numerical solution $\lambda_{\text{num}}(x, v)$ (left) and the exact solution $\lambda_{\text{exact}}(x, v)$ (right) over the computational domain.

For the numerical example, the regularization parameter is selected by the L-curve criterion; as shown in Figure 16, the corner identified via maximum discrete curvature yields $\alpha_{opt} = 4.6869 \times 10^{-1}$, which is used in the final reconstruction.

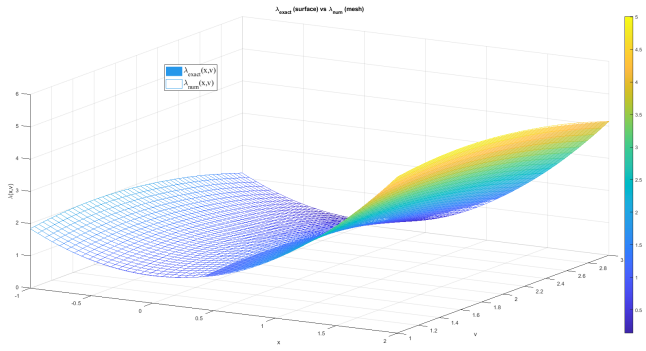
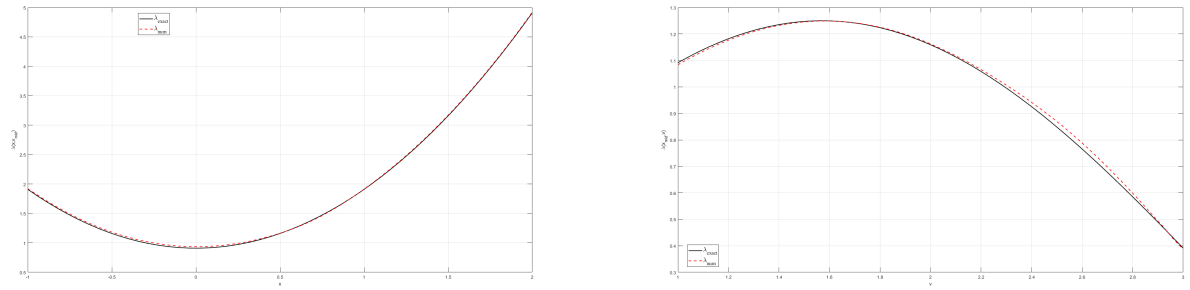
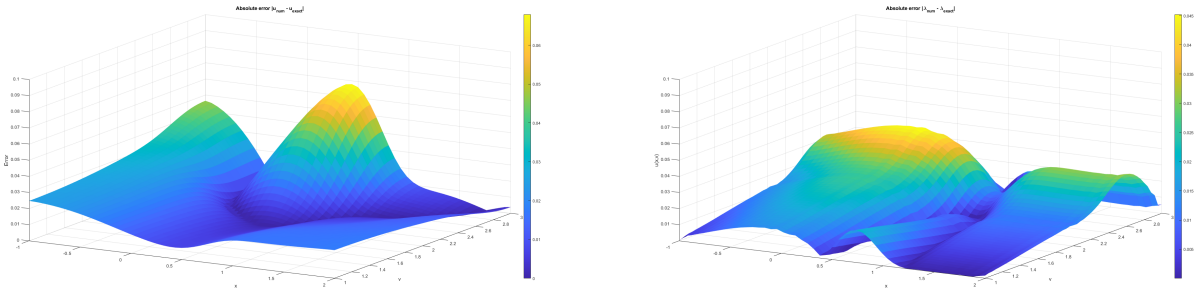


Figure 18. Comparison of the exact solution $\lambda_{\text{exact}}(x, v)$ (surface) and the reconstructed numerical solution $\lambda_{\text{num}}(x, v)$ (mesh) over the computational domain.



(a) (b)

Figure 19. Cross–sections of $\lambda(x, v)$: (a) cross–section for a fixed value of v; (b) cross–section for a fixed value of x, showing the comparison between the exact and numerical solutions.



(a) (b)

Figure 20. Absolute error distributions over the computational domain: (a) for the numerical solution u; (b) for the reconstructed source λ.

5. Conclusions

In this paper, we study an inverse problem for a stationary kinetic equation posed on a phase-space domain. The aim is to determine the function $u(x, v)$ together with an unknown source function $\lambda(x, v)$ from prescribed gradient data on the boundary and from the value of u at a single interior point of the domain. We develop a numerical algorithm based on a hybrid strategy that combines a finite-difference discretization with a bilinear interpolation polynomial, which results in a linear system for the numerical solution.

The numerical experiments show that the computational solution is in very good agreement with the exact solution. This is confirmed by surface and cross-sectional graphs and by the small values of absolute error over the entire computational domain. These results indicate that the proposed hybrid approach provides an accurate and computationally efficient tool for the numerical solution of this class of inverse kinetic problems.

While the present method demonstrates robust performance on rectangular grids using bilinear polynomials, its application to irregular geometries might lead to boundary-induced errors. As future work, we plan to extend this method to non-rectangular domains. This extension will involve adapting the framework to unstructured grids and exploring alternative spatial discretization techniques, such as finite element formulations, isoparametric mappings, or meshless methods, to preserve accuracy in complex geometries.

Building on these results, the proposed framework can be used as a basis for incorporating noisy or incomplete boundary data and for extending the analysis to higher-dimensional or time-dependent kinetic models.

Use of AI tools declaration

The authors used AI-assisted tools solely for grammar correction and language editing. All mathematical ideas, results, proofs, and scientific content are entirely the authors' original work.

Acknowledgments

The authors would like to thank the Editor and the anonymous reviewers for their careful reading and constructive comments, which helped improve the quality and clarity of this manuscript.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. C. Cercignani, *Theory and Application of the Boltzmann Equation*, Scottish Academic Press, 1975.
2. R. Liboff, *Introduction to the Theory of Kinetic Equations*, Wiley, 1969.
3. B. V. Alexeev, *Mathematical Kinetics of Reacting Gases*, Nauka, 1982.

4. C. Lancellotti, M. K. H. Kiessling, Self-similar gravitational collapse in stellar dynamics, *Astrophys. J.*, **549** (2001), L93–L96. https://doi.org/10.1086/319132
5. B. Perthame, Mathematical tools for kinetic equations, *Bull. Am. Math. Soc.*, **41** (2004), 205–244. https://doi.org/10.1090/S0273-0979-04-01004-3
6. P. A. Markowich, C. A. Ringhofer, C. Schmeiser, *Semiconductor Equations*, Springer, 1993.
7. C. Cercignani, G. A. Bird, E. Gabetta, *Transport Phenomena and Kinetic Theory: Applications to Gases, Semiconductors, Photons, and Biological Systems*, Springer Science & Business Media, 2007.
8. F. Natterer, *The Mathematics of Computerized Tomography*, Society for Industrial and Applied Mathematics, Wiley, 1986.
9. A. Kh. Amirov, *Integral Geometry and Inverse Problems for Kinetic Equations*, VSP, Utrecht, 2001. https://doi.org/10.1515/9783110940947
10. Y. E. Anikonov, *Inverse Problems for Kinetic and Other Evolution Equations*, VSP, Utrecht, 2001. https://doi.org/10.1515/9783110940909
11. A. Kh. Amirov, F. Gölgeleyen, Solvability of an inverse problem for the kinetic equation and a symbolic algorithm, *Comput. Model. Eng. Sci.*, **65** (2010), 179–190. https://doi.org/10.3970/cmes.2010.065.179
12. F. Gölgeleyen, A. Amirov, On the approximate solution of a coefficient inverse problem for the kinetic equation, *Math. Commun.*, **16** (2011), 283–298.
13. A. Kh. Amirov, F. Gölgeleyen, A. Rahmanova, An inverse problem for the general kinetic equation and a numerical method, *Comput. Model. Eng. Sci.*, **43** (2009), 131–147. https://doi.org/10.3970/cmes.2009.043.131
14. İ. Gölgeleyen, An inverse problem for a generalized transport equation in polar coordinates and numerical applications, *Inverse Probl.*, **29** (2013), 095006. https://doi.org/10.1088/0266-5611/29/9/095006
15. A. Amirov, Z. Ustaoğlu, B. Heydarov, Solvability of a two dimensional coefficient inverse problem for transport equation and a numerical method, *Transp. Theory Stat. Phys.*, **40** (2011), 1–22. https://doi.org/10.1080/00411450.2010.529980
16. A. Amirov, Z. Ustaoğlu, On the approximation methods for the solution of a coefficient inverse problem for a transport-like equation, *Comput. Model. Eng. Sci.*, **54** (2009), 283–299. https://doi.org/10.3970/cmes.2009.054.283
17. İ. Gölgeleyen, M. Hasdemir, A solution algorithm for an inverse source problem for the kinetic equation, *Int. J. Mod. Phys. C*, **33** (2022), 2250151. https://doi.org/10.1142/S0129183122501510
18. K. Yu, Z. Li, Y. Liu, Inverse source problem with a posteriori interior measurements for space-time fractional diffusion equations, *Comput. Math. Appl.*, **200** (2025), 305–314. https://doi.org/10.1016/j.camwa.2025.10.011
19. A. L. Karchevsky, L. A. Nazarova, L. A. Nazarov, Determining the fluid transport properties of a fractured porous reservoir by solving the inverse problem using well output measurements, *J. Appl. Mech. Tech. Phys.*, **62** (2021), 952–958. https://doi.org/10.1134/S0021894421060092

20. F. Gölgeleyen, İ. Gölgeleyen, M. Hasdemir, A hybrid solution method for an inverse problem for the general transport equation, *Comput. Math. Appl.*, **192** (2025), 172–188. https://doi.org/10.1016/j.camwa.2025.05.014

21. M. V. Klibanov, S. E. Pamyatnykh, Lipschitz stability of a non-standard problem for the non-stationary transport equation via a Carleman estimate, *Inverse Probl.*, **22** (2006), 881. https://doi.org/10.1088/0266-5611/22/3/009

22. M. V. Klibanov, S. E. Pamyatnykh, Global uniqueness for a coefficient inverse problem for the non-stationary transport equation via Carleman estimate, *J. Math. Anal. Appl.*, **343** (2008), 352–365. https://doi.org/10.1016/j.jmaa.2008.01.071

23. M. Machida, M. Yamamoto, Global Lipschitz stability in determining coefficients of the radiative transport equation, *Inverse Probl.*, **30** (2014), 035010. https://doi.org/10.1088/0266-5611/30/3/035010

24. A. Kawamoto, M. Machida, Global Lipschitz stability for a fractional inverse transport problem by Carleman estimates, *Appl. Anal.*, **100** (2021), 752–771. https://doi.org/10.1080/00036811.2019.1620931

25. P. Cannarsa, G. Floridia, F. Gölgeleyen, M. Yamamoto, Inverse coefficient problems for a transport equation by local Carleman estimate, *Inverse Probl.*, **35** (2019), 105013. https://doi.org/10.1088/1361-6420/ab1c69

26. M. V. Klibanov, J. Li, L. H. Nguyen, Z. Yang, Convexification numerical method for a coefficient inverse problem for the radiative transport equation, *SIAM J. Imaging Sci.*, **16** (2023), 35–63. https://doi.org/10.1137/22M1509837

27. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3^{rd} edition, Cambridge University Press, Cambridge, 2007.

28. J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, R. D. Phillips, *Computer Graphics: Principles and Practice*, 2^{nd} edition, Addison–Wesley, Reading (MA), 1996.

29. A. N. Tikhonov, V. Y. Arsenin, *Solutions of Ill-Posed Problems*, Winston, Washington (DC), 1977.

30. P. C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, *SIAM Rev.* **34** (1992), 561-580. https://doi.org/10.1137/1034115

31. P. C. Hansen, J. G. Nagy, D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Society for Industrial and Applied Mathematics, Philadelphia, 2006.

32. G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, Oxford, 1985.

Appendix

In this section, we present the structure of coefficient matrix $\tilde{A}$ in the matrix equation in (3.4). The matrices $\mathcal{A}_1^{(1)}$, $\mathcal{A}_2^{(i)}$, $\mathcal{A}_3^{(I)}$ are defined as follows:

$$\mathcal{A}_1^{(1)} = \begin{bmatrix} g_1(1,1) & g_2(1,1) & 0 & \cdots & 0 \\ g_3(1,2) & g_4(1,2) & g_5(1,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & g_3(1,J-1) & g_4(1,J-1) & g_5(1,J-1) \\ 0 & \cdots & 0 & \frac{2}{3}g_3(1,J) & g_6(1,J) \end{bmatrix}_{J\times J},$$

$$\mathcal{A}_2^{(i)} = \begin{bmatrix} h_1(i,1) & h_2(i,1) & 0 & \cdots & 0 \\ h_3(i,2) & h_4(i,2) & h_5(i,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & h_3(i,J-1) & h_4(i,J-1) & h_5(i,J-1) \\ 0 & \cdots & 0 & h_6(i,J) & h_7(i,J) \end{bmatrix}_{J\times J},$$

$$\mathcal{A}_3^{(I)} = \begin{bmatrix} m_1(I,1) & m_2(I,1) & 0 & \cdots & 0 \\ m_3(I,2) & m_4(I,2) & m_5(I,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & m_3(I,J-1) & m_4(I,J-1) & m_5(I,J-1) \\ 0 & \cdots & 0 & m_6(I,J) & m_7(I,J) \end{bmatrix}_{J\times J},$$

where

$g_1(i,j) = \left(\frac{8}{9}k_1 - \frac{8}{9}k_2 - \frac{2}{3}k_3 + \frac{2}{3}k_4 - \frac{4}{3}k_5 + \frac{4}{3}k_6\right)_{i,j}$, $g_2(i,j) = \left(-\frac{8}{9}k_1 + \frac{8}{9}k_2 - \frac{2}{3}k_4 - \frac{4}{3}k_6\right)_{i,j}$,

$g_3(i,j) = \left(\frac{2}{3}k_1 + \frac{4}{3}k_2 - k_4 + k_6\right)_{i,j}$, $g_4(i,j) = \left(-\frac{8}{3}k_2 - \frac{2}{3}k_3 + 2k_4 - \frac{4}{3}k_5\right)_{i,j}$,

$g_5(i,j) = \left(-\frac{2}{3}k_1 + \frac{4}{3}k_2 - k_4 - k_6\right)_{i,j}$, $g_6(i,j) = \left(-\frac{8}{9}k_1 - \frac{8}{9}k_2 - \frac{2}{3}k_3 + \frac{2}{3}k_4 - \frac{4}{3}k_5 - \frac{4}{3}k_6\right)$,

$h_1(i,j) = \left(\frac{8}{3}k_1 - 2k_3 + \frac{2}{3}k_4 + \frac{4}{3}k_6\right)_{i,j}$, $h_2(i,j) = \left(-\frac{8}{3}k_1 - \frac{2}{3}k_4 - \frac{4}{3}k_6\right)_{i,j}$,

$h_3(i,j) = (2k_1 + k_6 - k_4)_{i,j}$, $h_4(i,j) = (-2k_3 + 2k_4)_{i,j}$,

$h_5(i,j) = (-2k_1 - k_6 - k_4)_{i,j}$, $h_6(i,j) = \left(\frac{8}{3}k_1 - \frac{2}{3}k_4 + \frac{4}{3}k_6\right)_{i,j}$,

$h_7(i,j) = \left(\frac{2}{3}k_4 - 2k_3 - \frac{8}{3}k_1 - \frac{4}{3}k_6\right)_{i,j}$, $m_1(i,j) = \left(\frac{8}{9}k_1 + \frac{8}{9}k_2 - \frac{2}{3}k_3 + \frac{2}{3}k_4 + \frac{4}{3}k_5 + \frac{4}{3}k_6\right)_{i,j}$,

$m_2(i,j) = \left(-\frac{8}{9}k_1 - \frac{8}{9}k_2 - \frac{2}{3}k_4 - \frac{4}{3}k_6\right)_{i,j}$, $m_3(i,j) = \left(\frac{2}{3}k_1 - \frac{4}{3}k_2 - k_4 + k_6\right)_{i,j}$,

$m_4(i,j) = \left(\frac{8}{3}k_2 - \frac{2}{3}k_3 + 2k_4 + \frac{4}{3}k_5\right)_{i,j}$, $m_5(i,j) = \left(-\frac{2}{3}k_1 - \frac{4}{3}k_2 - k_4 - k_6\right)_{i,j}$,

$m_6(i,j) = \left(\frac{8}{9}k_1 - \frac{8}{9}k_2 - \frac{2}{3}k_4 + \frac{4}{3}k_6\right)_{i,j}$, $m_7(i,j) = \left(-\frac{8}{9}k_1 + \frac{8}{9}k_2 - \frac{2}{3}k_3 + \frac{2}{3}k_4 + \frac{4}{3}k_5 - \frac{4}{3}k_6\right)_{i,j}$.

In (3.4), the matrices $\mathcal{B}_1^{(1)}, \mathcal{B}_2^{(i)}, C_1^{(i)}, C_2^{(I)}$ are defined as follows:

$$\mathcal{B}_1^{(1)} = \begin{bmatrix} n_1(1,1) & n_2(1,1) & 0 & \cdots & 0 \\ n_3(1,2) & n_4(1,2) & n_5(1,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & n_3(1,J-1) & n_4(1,J-1) & n_5(1,J-1) \\ 0 & \cdots & 0 & n_6(1,J) & n_7(1,J) \end{bmatrix}_{J\times J},$$

$$\mathcal{B}_2^{(i)} = \begin{bmatrix} r_1(i,1) & -p_4(i,1) & 0 & \cdots & 0 \\ \frac{9}{8}n_6(i,2) & r_2(i,2) & \frac{9}{8}n_2(i,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \frac{9}{8}n_6(i,J-1) & r_2(i,J-1) & \frac{9}{8}n_2(i,J-1) \\ 0 & \cdots & 0 & -p_2(i,J) & r_3(i,J) \end{bmatrix}_{J\times J},$$

$$C_1^{(i)} = \begin{bmatrix} p_1(i,1) & p_2(i,1) & 0 & \cdots & 0 \\ -\frac{9}{8}n_2(i,2) & p_3(i,2) & -\frac{9}{8}n_6(i,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\frac{9}{8}n_2(i,J-1) & p_3(i,J-1) & -\frac{9}{8}n_6(i,J-1) \\ 0 & \cdots & 0 & p_4(i,J) & p_5(i,J) \end{bmatrix}_{J\times J},$$

$$C_2^{(I)} = \begin{bmatrix} q_1(I,1) & -\frac{9}{8}n_6(I,1) & 0 & \cdots & 0 \\ -n_5(I,2) & q_2(I,2) & -n_3(I,2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -n_5(I,J-1) & q_2(I,J-1) & -n_3(I,J-1) \\ 0 & \cdots & 0 & -n_2(I,J) & q_3(I,J) \end{bmatrix}_{J\times J},$$

where

$n_1(i,j) = \left(\frac{8}{9}k_2 - \frac{8}{9}k_1 + \frac{2}{3}k_3 + \frac{4}{3}k_5\right)_{i,j}$, $n_2(i,j) = \left(\frac{8}{9}k_1 - \frac{8}{9}k_2\right)_{i,j}$,

$n_3(i,j) = \left(-\frac{2}{3}k_1 - \frac{4}{3}k_2\right)_{i,j}$, $n_4(i,j) = \left(\frac{8}{3}k_2 + \frac{2}{3}k_3 + \frac{4}{3}k_5\right)_{i,j}$,

$n_5(i,j) = \left(\frac{2}{3}k_1 - \frac{4}{3}k_2\right)_{i,j}$, $n_6(i,j) = \left(-\frac{8}{9}k_1 - \frac{8}{9}k_2\right)_{i,j}$,

$n_7(i,j) = \left(\frac{8}{9}k_1 + \frac{8}{9}k_2 + \frac{2}{3}k_3 + \frac{4}{3}k_5\right)_{i,j}$, $p_1(i,j) = \left(-\frac{4}{3}k_1 - \frac{2}{3}k_2 + k_3 - k_5\right)_{i,j}$,

$p_2(i,j) = \left(\frac{4}{3}k_1 + \frac{2}{3}k_2\right)_{i,j}$, $p_3(i,j) = (-k_5 + k_3 - 2k_2)_{i,j}$,

$p_4(i,j) = \left(\frac{2}{3}k_2 - \frac{4}{3}k_1\right)_{i,j}$, $p_5(i,j) = \left(\frac{4}{3}k_1 - \frac{2}{3}k_2 + k_3 - k_5\right)_{i,j}$

$r_1(i,j) = \left(-\frac{4}{3}k_1 + \frac{2}{3}k_2 + k_3 + k_5\right)_{i,j}$, $r_2(i,j) = (k_5 + k_3 + 2k_2)_{i,j}$,

$r_3(i,j) = \left(\frac{4}{3}k_1 + \frac{2}{3}k_2 + k_3 + k_5\right)_{i,j}$, $q_1(i,j) = \left(\frac{2}{3}k_3 - \frac{8}{9}k_2 - \frac{8}{9}k_1 - \frac{4}{3}k_5\right)_{i,j}$,

$q_2(i,j) = \left(\frac{2}{3}k_3 - \frac{8}{3}k_2 - \frac{4}{3}k_5\right)_{i,j}$, $q_3(i,j) = \left(\frac{8}{9}k_1 - \frac{8}{9}k_2 + \frac{2}{3}k_3 - \frac{4}{3}k_5\right)_{i,j}$.

AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)