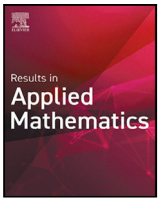
Results in Applied Mathematics 29 (2026) 100686

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics



On the inversion of polynomials of discrete Laplace matrices

S. Asghar [a,b,*], Q. Peng [c,d], F.J. Vermolen [a,b,e,f], C. Vuik [e]

[a] Computational Mathematics Group (CMAT), Department of Mathematics and Statistics, University of Hasselt, Diepenbeek, Belgium
[b] Data Science Institute (DSI), University of Hasselt, Diepenbeek, Belgium
[c] Mathematics for AI in Real-world Systems, School of Mathematical Sciences, Lancaster University, Lancaster, UK
[d] Mathematical Institute, Leiden University, Leiden, The Netherlands
[e] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
[f] Department of Mathematics and Applied Mathematics, University of Johannesburg, Johannesburg, South Africa

ARTICLE INFO

Keywords:
Laplace matrix
Inverse matrix
Solution to linear systems
Eigenvector expansion

ABSTRACT

The efficient inversion of matrix polynomials is a critical challenge in computational mathematics. We design a procedure to determine the inverse of matrices polynomial of multidimensional Laplace matrices. The method is based on eigenvector and eigenvalue expansions. The method is consistent with previously known expressions of the inverse discretized Laplacian in one spatial dimension (Vermolen et al., 2022). The formalism is further extended to obtain closed form expressions for time-dependent problems.

1. Introduction

The inversion of discretized Laplace inspired matrices is often a crucial, but also rate-determining step in many simulation packages [1–3]. Such simulations may come from diffusion, heat distribution, fluid dynamics and many other modeling problems from science and technology. Even from financial mathematics or filtering in statistics and data science, such problems are important to solve [4,5]. Developing efficient solvers and pre-conditioners for these matrices remains a core focus in numerical computing to improve simulation efficiency and accuracy.

The current paper addresses a solution to the problem

$$A\underline{x} = \underline{b},$$

where $A \in \mathbb{R}^{n\times n}$, where $n \in \mathbb{N}$ is relatively large. Solutions of large linear algebraic systems of equations often proceed by the use of Krylov subspace methods, multigrid methods or combinations of the two. In the case of very large systems, parallelized algorithms may be beneficial. Such systems often result from the discretization of a partial differential equations (PDEs) based on a Laplacian operator, such as

$$-\Delta u = -\nabla\cdot(\nabla u) = f(\mathbf{x}), \text{ in } \Omega,$$

where $\Omega \subsetneq \mathbb{R}^d$ is an open, connected, bounded domain, and appropriate boundary conditions are assigned to u that warrant existence and uniqueness of the solution. In these cases, the discrete counterpart of the differential operator, being the matrix A, should be invertible if an adequate discretization (in terms of consistency, stability and hence convergence) is used. Some other PDEs, such as the biharmonic equation or the Cahn–Hilliard equation, contain (linear) combinations of powers of Laplace operators. We will consider such types of equations. The resolution of these types of equations is of significant interest due to its broad applicability

* Corresponding author.
E-mail address: sabia.asghar@uhasselt.be (S. Asghar).

https://doi.org/10.1016/j.rinam.2026.100686
Received 26 November 2025; Received in revised form 19 January 2026; Accepted 27 January 2026
Available online 28 January 2026

across multiple disciplines, including physics, chemistry, mathematics, economics [6], and machine learning/AI [7,8]. Their general forms encompass several fundamental classes of partial differential equations (PDEs), each governing critical phenomena:

- Elliptic Equations: The Poisson equation, which models electrostatic and magnetostatic fields, incompressible fluid flow (both inviscid and viscous), vortex dynamics, and fluid filtration through porous media. Somewhat more complicated, but in the same philosophy, we mention (linear) elasticity as an important application in mechanics, and the Euler–Tricomi equation, which is essential for studying transonic flow dynamics.
- Parabolic Equations: The extended diffusion equation and the Diffusion–Advection–Reaction (DAR) equation, which describe processes such as heat transfer, mass transport, and electromagnetic field propagation. The Black–Scholes equation, widely used in mathematical finance for option pricing [9].
- Hyperbolic Equations: The extended wave equation and telegraph equation, which play a crucial role in electromagnetics and telecommunications [10]. Other examples concerns Buckley–Leverett or Burgers' equations that arise in applications in modeling flow in porous media.

Given its wide-ranging implications, we propose a simple, accurate, and possibly computationally efficient numerical approach that aims to provide an exact solution while maintaining high precision, striking an optimal balance between simplicity and accuracy. Furthermore, the approach that we use to obtain closed form solutions to a class of systems of linear equations, as well as inverses of classes of matrices is of theoretical value. Convection-based operators, or equations based on odd-order spatial derivatives, as well as nonlinear problems, will not be considered in this paper. Furthermore, we will limit ourselves to linear combinations of powers of Laplace operators. We will develop a closed-form expression for the solution $\underline{x}$, as well as a formalization to express A^{-1}. One usually does not determine the inverse of a matrix, however, our method allows to do so for theoretical purposes. The analysis and approach will be conceptually remarkably simple since it is based on Von Neumann analysis, which we will not use for the assessment of stability, but for the sake of determining eigenvalues of A.

The paper deals with the important class of discrete Laplace matrices. In Section 2, we introduce a numerically efficient method based on eigenvector expansions and derive the underlying inversion principle in Section 3. The practical utility and performance of our technique are then validated through a series of case studies in Section 4, followed by discussion and conclusions in Section 5.

2. The eigenvector expansion technique

Consider the linear system of equations

$$A\underline{x} = \underline{b}, \tag{1}$$

where A is a nonsingular, symmetric $n \times n$ matrix and $\underline{x}$ and $\underline{b}$ are vectors in $\mathbb{R}^n$. The matrix A has n orthogonal eigenvectors by the Principal Axis Theorem [11] for symmetric matrices, which form a basis for $\mathbb{R}^n$. In particular, there is a set of n eigenvectors $\{\underline{v}_k\}$, and a corresponding set of real eigenvalues $\{\lambda_k\}$, such that

$$A\underline{v}_k = \lambda_k \underline{v}_k, \quad k = 1, 2, \ldots, n \tag{2}$$

and

$$(\underline{v}_i, \underline{v}_j) = 0 \quad \text{if } i \neq j,$$

where $(\cdot, \cdot)$ is the standard vector inner (dot) product

$$(\underline{x}, \underline{y}) = \frac{1}{n} \sum_{k=1}^{n} x_k y_k.$$

Since the set of vectors $\{\underline{v}_k\}$ forms a basis, one can express $\underline{b}$ as a linear combination of those vectors

$$\underline{b} = \sum_{k=1}^{n} \beta_k \underline{v}_k.$$

As the set of vectors is orthogonal, one can determine the coefficients β_k by forming inner products of Eq. (2) with each of the elements of the set $\{\underline{v}_k\}$. For example, to obtain an expression for β_j, we form the inner product of Eq. (2) with $\underline{v}_j$ and simplify

$$(\underline{b}, \underline{v}_j) = \left(\sum_{k=1}^{n} \beta_k \underline{v}_k, \underline{v}_j \right) = \sum_{k=1}^{n} \beta_k (\underline{v}_k, \underline{v}_j) = \beta_j (\underline{v}_j, \underline{v}_j).$$

Hence,

$$\beta_j = \frac{(\underline{b}, \underline{v}_j)}{(\underline{v}_j, \underline{v}_j)}. \tag{3}$$

Note that we did not yet normalize the eigenvectors of A. We can express the solution $\underline{x}$ of (1) as a linear combination of the vectors $\{\underline{v}_k\}$

$$\underline{x} = \sum_{k=1}^{n} c_k \underline{v}_k.$$

If $\underline{x}$ is to be a solution of $A\underline{x} = \underline{b}$, then

$$A\left(\sum_{k=1}^{n} c_k \underline{v}_k\right) = \underline{b}.$$

Using the eigenvector property of $\underline{v}_k$, one gets

$$\sum_{k=1}^{n} c_k A\underline{v}_k = \sum_{k=1}^{n} \beta_k \underline{v}_k$$

$$\Rightarrow \sum_{k=1}^{n} c_k \lambda_k \underline{v}_k = \sum_{k=1}^{n} \beta_k \underline{v}_k.$$

By forming inner products with $\underline{v}_j$, we obtain

$$\sum_{k=1}^{n} c_k \lambda_k (\underline{v}_k, \underline{v}_j) = \sum_{k=1}^{n} \beta_k (\underline{v}_k, \underline{v}_j).$$

Using orthogonality,

$$c_j \lambda_j (\underline{v}_j, \underline{v}_j) = \beta_j (\underline{v}_j, \underline{v}_j).$$

Thus

$$c_j = \frac{\beta_j}{\lambda_j}.$$

Hence we conclude that, if the matrix A possesses a set of n orthogonal eigenvectors, then the solution to $A\underline{x} = \underline{b}$ is given by the eigenvector expansion

$$\underline{x} = \sum_{k=1}^{n} \frac{\beta_k}{\lambda_k} \underline{v}_k,$$

where $\lambda_k \neq 0$ is the eigenvalue associated with $\underline{v}_k$, and the β_k's are determined from Eq. (3). The above relation is valid if the matrix A is diagonalizable and if the matrix has n orthonormal eigenvectors. This holds for real, symmetric or Hermitian matrices. However, in many cases, there is no explicit relation for the eigenvalues and eigenvectors of the matrix, which reduces the applicability of this formula. In the case of Laplace matrices, we will take advantage of the explicit forms of the eigenvalues and eigenvectors. Note that this procedure is analogous to the procedures used in separation of variables in solving partial differential equations and that we did not yet normalize the eigenvectors.

3. The principle for inverses of matrix polynomials

We continue with the system $A\underline{x} = \underline{b}$, where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix (for instance representing the finite difference representation of the Laplace operator with Dirichlet boundary conditions), which gives positive (real-valued) eigenvalues and orthogonal eigenvectors. Let the normalized eigenvectors of A be given by $\underline{v}_k$, $k = 1, \ldots, n$, with respective eigenvalues λ_k, then we arrive at the following expression for the solution

$$\underline{x} = \sum_{j=1}^{n} x_j \underline{v}_j = \sum_{j=1}^{n} \frac{1}{\lambda_j} (\underline{b}, \underline{v}_j) \underline{v}_j. \tag{4}$$

This entirely fits within the eigenvalue expansion in the continuous case. The next step is to express the inverse A^{-1} in terms of the eigenvalues and eigenvectors. In determining the inverse, one can proceed columnwisely. Let $\underline{g}_k$ be the kth column of A^{-1}, then $\underline{g}_k$ satisfies

$$A\underline{g}_k = \underline{e}_k = [\ldots 1 \ldots]^T. \tag{5}$$

Here $\underline{e}_k$ is a vector with zeros, except for the kth position where it has the value 1. Using Eq. (4), this implies that

$$\underline{g}_k = \sum_{j=1}^{n} \frac{1}{\lambda_j} (\underline{e}_k, \underline{v}_j) \underline{v}_j = \sum_{j=1}^{n} \frac{v_{jk}}{n\lambda_j} \underline{v}_j, \tag{6}$$

where v_{jk} represents the kth component of the jth eigenvector of A. Note that we used $\underline{e}_k$ and the division by n for the inner product. In order to get element $(A^{-1})_{ik}$, one computes from

$$(A^{-1})_{ik} = G_{ki} = \sum_{j=1}^{n} \frac{v_{jk}}{n\lambda_j} v_{ji}, \tag{7}$$

where $G = [\underline{g}_1 \ldots \underline{g}_n]$ is the matrix with the g-vectors as its columns. Hence, formally $A^{-1} = G^T$. Note that A^{-1} is symmetrical since the inverse of a symmetric positive definite matrix is also symmetric positive definite, and therefore we have $A^{-1} = G$. The principle

of the above equation is used in the computations that will follow in the later sections. Furthermore, we note that if λ_j and $\underline{v}_j$ form an eigenpair of the symmetric positive definite matrix A, then the matrix A^m, $m \in \mathbb{Z}$, is also symmetric positive definite with eigenpair λ_j^m and $\underline{v}_j$. Hence the solution to

$$A^m \underline{x} = \underline{b}, \tag{8}$$

is given by

$$\underline{x} = \sum_{j=1}^{n} \frac{1}{\lambda_j^m} (\underline{b}, \underline{v}_j) \underline{v}_j, \tag{9}$$

and the inverse of A^m, denoted by A^{-m}, is given by

$$(A^{-m})_{ik} = \sum_{j=1}^{n} \frac{v_{jk}}{n\lambda_j^m} \, v_{ji}. \tag{10}$$

This is generalized in the following assertion:

Theorem 1. *Let A be an $n \times n$ matrix over $\mathbb{R}$ with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ (counting algebraic multiplicities). For any polynomial*

$$P_m(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0,$$

the matrix $P_m(A)$ has eigenvalues $P_m(\lambda_i)$ for $i = 1, \ldots, n$.

Proof. For the proof, the reader is referred to [12,13]. □

Corollary 1 (*Extension to Diagonalizable Matrices*)**.** *If A is diagonalizable with $A = QDQ^{-1}$, then*

$$P_m(A) = Q \begin{pmatrix} P_m(\lambda_1) & & \\ & \ddots & \\ & & P_m(\lambda_n) \end{pmatrix} Q^{-1}.$$

Subsequently, knowing the eigenvalues and eigenvectors of A, then for the general matrix polynomial of order m, one obtains for each eigenpair λ and $\underline{v}$

$$P_m(A)\underline{v} = (\beta_0 I + \beta_1 A + \cdots + \beta_m A^m)\underline{v} = (\beta_0 + \beta_1 \lambda + \cdots + \beta_m \lambda^m)\underline{v}, \tag{11}$$

which implies that this matrix polynomial has the same eigenvectors as A with eigenvalue $P_m(\lambda) = \beta_0 + \beta_1 \lambda + \cdots + \beta_m \lambda^m$, which according to the Fundamental Theorem of Algebra [14], has at least one (complex) zero and the polynomial can be factorized in terms of its zeros by the Factorization Theorem. If $\beta_j > 0, \forall\, j$, then this expression will never be zero since $\lambda > 0$ (recall that A is symmetric positive definite). The matrix polynomial is symmetric positive definite. Therefore, for the equation

$$P_m(A)\underline{x} = \underline{b}, \tag{12}$$

the solution is expressed by

$$\underline{x} = \sum_{j=1}^{n} \frac{1}{P_m(\lambda_j)} (\underline{b}, \underline{v}_j) \underline{v}_j. \tag{13}$$

Hence for the inverse of a matrix polynomial, we get

$$(P_m(A)^{-1})_{ik} = \sum_{j=1}^{n} \frac{v_{jk}}{nP_m(\lambda_j)} \, v_{ji}. \tag{14}$$

Hence we have expressed the solution to a linear system of equations and the inverse of a matrix, in case that the matrix is polynomial in a symmetric positive definite matrix, A, in terms of the eigenvalues and eigenvectors of the original matrix A. Once the eigenvalues and eigenvectors of A have been determined, then it is straightforward to determine the inverse of any polynomial provided that the matrix $P_m(A)$ is nonsingular. Of course we note that in most cases one is not interested in the inverse of the matrix polynomial, but merely in the solution $\underline{x}$. In the next section, we will derive some practical results.

4. Case studies

4.1. One dimensional Laplacian matrix

We first illustrate how the method works for a one-dimensional case and we show that our result is consistent with our earlier results [15]. For this purpose, we consider a simple Dirichlet problem, given by

$$\begin{cases} -u'' = f(x), \quad 0 < x < 1, \\ u(0) = u(1) = 0. \end{cases}$$

We divide the interval $(0,1)$ into equidistant mesh points, $x_j = jh$ with spacing h. Having n unknowns, we have $(n+1)h = 1$. Note that the total number of nodal points is $n+2$ and that the number of degrees of freedom (unknowns) is given by n. Having Dirichlet boundary conditions, gives the $n \times n$-matrix A with entries $a_{ij} \in \mathbb{R}^{n\times n}$ are given by

$$a_{ij} = \frac{1}{h^2} \cdot \begin{cases} 2, & \text{if } i = j, \\ -1, & \text{if } |i-j| = 1, \\ 0, & \text{otherwise,} \end{cases} \qquad i,j = 1,\dots,n. \tag{15}$$

or can be expressed as

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}. \tag{16}$$

The above matrix results from elimination of the Dirichlet boundary conditions, and is symmetric positive definite. In our previous work [15], the solution to the equation $A\underline{x} = \underline{b}$ was obtained through the mapping of the piecewise linear fundamental solution to the continuous problem to the finite difference mesh. Since the higher-order derivatives of the fundamental solution are zero, the exact solution and numerical solution are equal. This leads to an expression for A^{-1}. This has been done for generic meshes and boundary conditions in $\mathbb{R}^1$. Now we use an approach based on the eigenvectors and eigenvalues of this matrix A. The corresponding continuous eigenvalue problem becomes

$$\begin{cases} u'' + \mu^2 u = 0, \\ u(0) = u(1) = 0. \end{cases} \tag{17}$$

The eigenvectors and (normalized) eigenfunctions of this Sturm–Liouville problem are given by

$$\hat{\lambda}_j = \mu_j^2 = j^2\pi^2, \qquad \phi_j(x) = \sqrt{2}\sin(j\pi x), \qquad j = 1,2,\dots. \tag{18}$$

As an Ansatz for the eigenvectors of A, we project the eigenfunction on the finite difference mesh $\{x_j\}_{j=1}^n$, with $x_j = jh$ and $h = \frac{1}{n+1}$. Then for the ith row multiplication of A, one obtains

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}\phi_k(x_j) &= \frac{1}{h^2}(-\phi_k(x_j - h) + 2\phi_k(x_j) - \phi_k(x_j + h)) \\ &= \frac{2\sqrt{2}}{h^2}(1 - \cos(k\pi h))\sin(k\pi x_j) = \lambda_k \phi_k(x_j). \end{aligned}$$

Hence the projection of the eigenfunctions of the continuous problem onto the finite difference mesh points gives the eigenvectors of the discretization matrix. Furthermore, the eigenvalues of the discretization matrix are given by

$$\lambda_k = \frac{2}{h^2}(1 - \cos(k\pi h)) = \frac{4}{h^2}\sin^2\left(\frac{k\pi h}{2}\right) \in \left(0, \frac{4}{h^2}\right), \text{ for } k = 1,\dots,n, \tag{19}$$

with corresponding eigenvector

$$v_{kj} = \sqrt{2}\sin(k\pi x_j) = \sqrt{2}\sin(k\pi jh) = \phi_k(x_j) = \phi_k(jh), \qquad j = 1,\dots,n. \tag{20}$$

Hence the eigenvectors are indeed constructed from the projection of the eigenfunctions of the continuous Laplace operator onto the finite difference mesh. Then the eigenvalues and eigenvectors are available and can be used to compute the solution to $A\underline{x} = \underline{b}$ and the inverse A^{-1}, as well as the inverse of a matrix polynomial. Using Eq. (4) for the current one-dimensional Laplace operator with Dirichlet boundary conditions, the solution of $A\underline{x} = \underline{b}$ is formally given by

$$\underline{x} = \sum_{j=1}^{n} \frac{h^2(\underline{b}, \underline{v}_j)}{4\sin^2(\frac{j\pi h}{2})}\underline{v}_j = \frac{h^2}{n+1}\sum_{j=1}^{n} \frac{\sum_{k=1}^{n+1}(b_k v_{jk})}{4\sin^2(\frac{j\pi h}{2})}\underline{v}_j, \tag{21}$$

where $v_{jk} = \sqrt{2}\sin(j\pi kh)$ has been chosen. Further for the inner product, $n+1$ has been chosen since the number of intervals $x_j - x_{j-1} = h$ is $n+1$. This gives consistency with 'continuous inner product' based on an integral. Note further that $v_{j,n+1} = 0$, which implies that the $n+1$-th term does not have any influence and the only adaptation is the division by $n+1$ instead of n for the sake of consistency with the $n+1$ intervals. Hence the inverse of A is expressed by

$$(A^{-1})_{ik} = G_{ki} = \frac{h^2}{4(n+1)}\sum_{j=1}^{n} \frac{v_{jk}}{\sin^2(\frac{j\pi h}{2})}\, v_{ji} = \frac{h^2}{2(n+1)}\sum_{j=1}^{n} \frac{\sin(j\pi kh)}{\sin^2(\frac{j\pi h}{2})}\, \sin(j\pi ih). \tag{22}$$

This principle can be used for different boundary conditions as long as we do not have Neumann conditions on both boundaries because of singularity of A (then A^{-1} does not exist at all, so it becomes pointless). However, double Neumann conditions can be used for matrix polynomials. For the one-dimensional case, this new approach is of hardly any value compared to the fundamental

solution approach which is useful since the fundamental solutions are piecewise linear in $\mathbb{R}^1$, which makes the truncation errors zero under the use of finite differences or finite elements as a result of the second derivative being zero between adjacent meshpoints.

For the 1D Laplace equation with Dirichlet boundary conditions, the inverse matrix is given by

$$(A^{-1})_{ik} = h^2 \left(\frac{n+1-k}{n+1} i - (i-k)_+ \right), \tag{23}$$

which is a much simpler expression (see for instance [15]). However, for higher dimensionality, the fundamental solution is typically a logarithmic function, which has global nonzero higher-order partial derivatives, and hence the truncation error will not vanish. The same holds for polynomials of Laplace operators, even in 1D, such as the relatively simple case $-u'' + u$. For this reason, the fundamental solutions are no longer useful for both higher-dimensional problems and problems, which entail polynomials of the Laplace operator. It is easy to verify, by means of implementing both formulas in the computer, that these different expressions are equal. However, it is more elegant to verify this rigorously from a mathematical point of view using discrete Fourier transforms. We will summarize the result as a theorem and prove the equality.

Theorem 2. *Let A be given by Eq. (16), then its inverse is given by*

$$(A^{-1})_{ik} = h^2 \left(\frac{n+1-k}{n+1} \, i - (i-k)_+ \right) = \frac{h^2}{2(n+1)} \sum_{j=1}^{n} \frac{\sin(j\pi kh)}{\sin^2(\frac{j\pi h}{2})} \, \sin(j\pi ih).$$

Proof. We remark that we just computed the right-hand side and that the first equality was determined in [15]. Hence, these expressions should give the same value. However, because these expressions look very different, we will algebraically demonstrate that they are consistent. Since the functions $\phi_k(x) = \sqrt{2}\sin(k\pi x)$ are eigenfunctions and eigenvectors (using $v_j = \phi_k(x_j)$) of the Laplace and Laplace matrix, respectively. Since the Laplace matrix is symmetric, the eigenvectors are orthogonal (in the continuous case, this holds as well, of course). Hence, the functions satisfy

$$\sum_{j=1}^{n} \phi_k(x_j)\phi_l(x_j) = 0, \quad \text{if } k \neq l.$$

We can normalize the functions by choosing α in $\phi_k(x_j) = \alpha \, \sin(k\pi jh)$ so that

$$\sum_{j=1}^{n} \alpha^2 \sin^2(k\pi jh) = 1.$$

To this extent, we compute

$$\begin{aligned}
\sum_{j=1}^{n} \sin^2(k\pi jh) &= \sum_{j=1}^{n} \left(\frac{1}{2} - \frac{1}{2}\cos(2k\pi jh) \right) \\
&= \frac{n}{2} - \frac{1}{4} \sum_{j=1}^{n} \left(e^{2k\pi jhi} + e^{-2k\pi jhi} \right) \\
&= \frac{n}{2} - \frac{1}{4} \sum_{j=1}^{n} (e^{2k\pi hi})^j - \frac{1}{4} \sum_{j=1}^{n} (e^{-2k\pi hi})^j \\
&= \frac{n}{2} - \frac{1}{4} \left(\frac{1 - e^{2k\pi(n+1)hi}}{1 - e^{2k\pi hi}} - 1 + \frac{1 - e^{-2k\pi(n+1)hi}}{1 - e^{-2k\pi hi}} - 1 \right) \\
&= \frac{n+1}{2},
\end{aligned}$$

where we used $(n+1)h = 1$, and the fact that the complex exponential is one if the argument is any integer times 2π. This implies that we have $\alpha = \sqrt{\frac{2}{n+1}}$, hence in the discrete setting, we have the following normalized eigenvectors

$$v_{kj} = \phi_k(x_j) = \sqrt{\frac{2}{n+1}} \sin(k\pi jh).$$

For a general discrete function, we can write the discrete sine Fourier transform by

$$f(x_i) = \sum_{k=1}^{n} c_k \phi_k(x_i),$$

using normalized functions, we have

$$c_k = \sum_{j=1}^{n} f(x_j)\phi_k(x_j).$$

Note that Eq. (23) can be written as

$$(A^{-1})_{ik} = h^2 \left(\frac{n+1-k}{n+1} x_i - (x_i - x_k)_+ \right) = \sum_{l=1}^{n} c_l \phi_l(x_i).$$

Inserting $\phi_l(x) = \sqrt{\frac{2}{n+1}}\sin(l\pi x)$, gives

$$c_l = \sum_{j=1}^{n}(A^{-1})_{jk}\phi_l(x_j) = \sqrt{\frac{2}{n+1}}\sum_{j=1}^{n} h^2\left(\frac{n+1-k}{n+1}j - (j-k)_+\right)\sin(l\pi jh).$$

We use the relation

$$s_n(z) = \sum_{k=0}^{n} kz^k = \sum_{k=1}^{n} kz^k = \begin{cases} \frac{1}{2}n(n+1), & \text{if } z = 1, \\ \dfrac{z(1-(n+1)z^n + nz^{n+1})}{(1-z)^2}, & \text{if } z \neq 1. \end{cases} \tag{24}$$

This relation can be demonstrated by mathematical induction or by differentiation and subsequent multiplication by z of the geometric series. By writing sines and cosines as linear combinations of complex exponentials, we can use the above relation to arrive at

$$\sum_{k=1}^{n} k\ \sin(kx) = \frac{(n+1)\sin(nx) - n\sin((n+1)x)}{4\sin^2(\frac{x}{2})}, \quad x \neq 2p\pi,\ p \in \mathbb{Z}, \tag{25}$$

$$\sum_{k=1}^{n} k\ \cos(kx) = \frac{(n+1)\cos(nx) - n\cos((n+1)x) - 1}{4\sin^2(\frac{x}{2})}, \quad x \neq 2p\pi,\ p \in \mathbb{Z}. \tag{26}$$

The above relation is used to write

$$\sum_{j=1}^{n}(j-k)_+\sin(jl\pi h) = \sum_{j=k+1}^{n}(j-k)\sin(jl\pi h) = \sum_{p=1}^{n-k} p\ \sin((p+k)l\pi h). \tag{27}$$

This is further worked out by

$$\begin{aligned}
\sum_{p=1}^{n-k} p\sin((p+k)l\pi h) &= \sin(kl\pi h)\sum_{p=1}^{n-k} p\cos(pl\pi h) + \cos(kl\pi h)\sum_{p=1}^{n-k} p\sin(pl\pi h) \\
&= \sin(kl\pi h)\frac{(n-k+1)\cos((n-k)l\pi h) - (n-k)\cos((n-k+1)l\pi h) - 1}{4\sin^2(\frac{l\pi h}{2})} \\
&\quad + \cos(kl\pi h)\frac{(n-k+1)\sin((n-k)l\pi h) - (n-k)\sin((n-k+1)l\pi h)}{4\sin^2(\frac{l\pi h}{2})} \\
&= \frac{(n-k+1)\sin(nl\pi h) - (n-k)\sin(l(n+1)\pi h) - \sin(kl\pi h)}{4\sin^2(\frac{l\pi h}{2})} \\
&= \frac{(n-k+1)\sin(nl\pi h) - \sin(kl\pi h)}{4\sin^2(\frac{l\pi h}{2})}.
\end{aligned}$$

Here we used $\sin(x+y) = \sin x\cos y + \cos x\sin y$, $(n+1)h = 1$ and the fact that the sine is zero in arguments that are multiples of π. Hence, for c_l, we obtain

$$\begin{aligned}
c_l &= \sqrt{\frac{2}{n+1}}h^2\left[\frac{n+1-k}{n+1}\frac{(n+1)\sin(nl\pi h)}{4\sin^2(\frac{l\pi h}{2})} - \frac{(n-k+1)\sin(nl\pi h) - \sin(kl\pi h)}{4\sin^2(\frac{l\pi h}{2})}\right] \\
&= \sqrt{\frac{2}{n+1}}h^2\frac{\sin(kl\pi h)}{4\sin^2(\frac{l\pi h}{2})}.
\end{aligned}$$

Hence, this gives

$$(A^{-1})_{ik} = \sum_{l=1}^{n} c_l\sqrt{\frac{2}{n+1}}\sin(il\pi h) = \frac{h^2}{2(n+1)}\sum_{l=1}^{n}\frac{\sin(kl\pi h)}{\sin^2(\frac{l\pi h}{2})}\sin(il\pi h). \quad \square$$

By this we have demonstrated consistence of the current result with the earlier [15], simpler result for one spatial dimension with Dirichlet boundary conditions. Consistency for different boundary conditions can be approached using similar principles, although the algebra may be a little more tedious. We therefore omit this at this stage. Our new approach gives the exact inverse of the matrix represented by a discrete Fourier transform. We realize that the new approach to invert the one-dimensional Laplace matrix is much more toilsome than the earlier procedure. However, computing the solution from a polynomial matrix equation was not possible with the earlier method, whereas the current method can be used to compute the solution to this problem. For the sake of illustration, we solve the following problem

$$\begin{cases} -u'' + \alpha u = 1, \text{ for } x \in (0,1), \\ u(0) = u(1) = 0. \end{cases}$$

The solution can be seen in Fig. 1 for different values of $\alpha \geq 0$. For $\alpha > 0$, the simple formula previously found for the inverse [15] cannot be used since the fundamental solution is composed by hyperbolic sines and cosines and hence contains non-zero higher-order

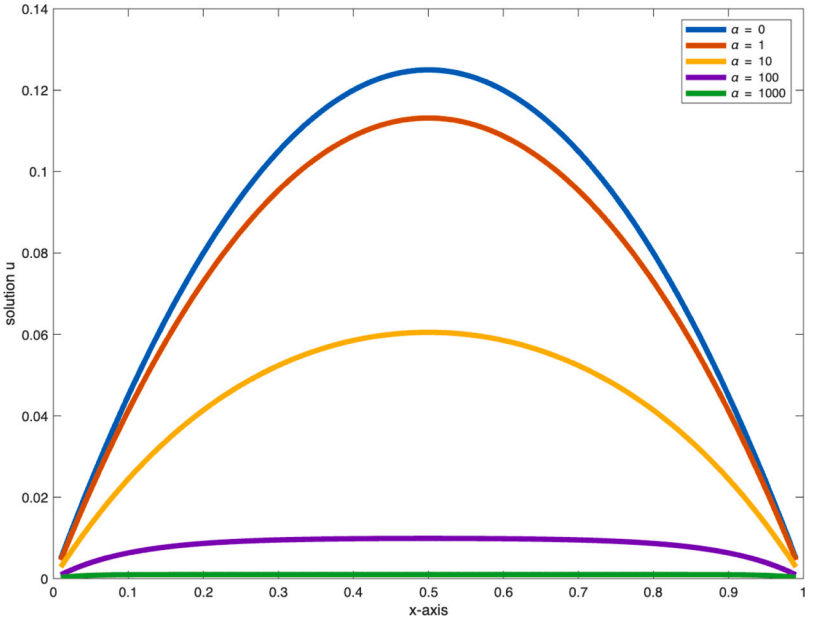


Fig. 1. Solutions to the equation $-u'' + \alpha u = 1$ for different α-values $\alpha = 0, 1, 10, 100, 1000$.

derivatives, which gives a nonzero truncation error. In [15], an approximation of the inverse based on hyperbolic sines and cosines with error $\mathcal{O}(h^{3/2})$ has been derived.

4.2. Two dimensional Laplace on a rectangle

Gueye [16] gave an exact inversion of a symmetric pentadiagonal Toeplitz matrix for the semi-analytic solution of 2D Poisson equation. For this case, no explicit relation for the inverse of the discretization matrix is known to the best of our knowledge. We consider a unit square with N unknowns at equidistant spacing h in both directions. Hence there are $n = N^2$ unknowns in total. We consider the simple Laplacian with homogeneous Dirichlet conditions, which typically reads as

$$\cdot \begin{cases} -\Delta u = f(x,y), \text{ for } (x,y) \in \Omega = (0,1)^2, \\ u|_{\partial\Omega} = 0. \end{cases}$$

The corresponding eigenvalue problem is given by

$$-\Delta\hat{\phi}_{j_1,j_2} = \hat{\lambda}_{j_1,j_2}\hat{\phi}_{j_1,j_2} = \hat{\mu}^2_{j_1,j_2}\hat{\phi}_{j_1,j_2} = (\hat{\mu}^2_{j_1} + \hat{\mu}^2_{j_2})\hat{\phi}_{j_1,j_2},$$

where $j_1, j_2 \in \mathbb{N}^{\times} = \mathbb{N} \setminus \{0\}$, and $\hat{\mu}^2_{j_p} = \pi^2 j_p^2$, hence for the eigenvalues, we have

$$\hat{\lambda}_{k_1,k_2} = \hat{\mu}^2_{k_1,k_2} = \pi^2(k_1^2 + k_2^2), \qquad k_j = 1, 2, 3, \dots,$$

and for the eigenfunctions we have

$$\hat{\phi}_{j_1,j_2}(x,y) = 2\sin(\pi j_1 x)\sin(\pi j_2 y).$$

The 2D finite difference approach with constant spacing and N unknowns per spatial dimension, gives the following discretization matrix (shown for $N = 4$)

$$A = \frac{1}{h^2} \cdot \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & \dots \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & \dots \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 & \dots \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 & \dots \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 & \dots \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 & \dots \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 & \dots \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 & \dots \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix},$$

with the following eigenvalues for the discrete system

$$\lambda_{k_1,k_2} = \frac{4}{h^2}\left(\sin^2(\frac{k_1\pi h}{2}) + \sin^2(\frac{k_2\pi h}{2})\right), \qquad \text{for } k_j \in \{1,\dots,N\}, \tag{28}$$

with eigenvectors

$$v_{\hat{j},\hat{k}} = w_{(j_1,j_2),(k_1,k_2)} = 2\ \sin(j_1\pi k_1 h)\ \sin(j_2\pi k_2 h), \qquad \text{for } j_m,\ k_m \in \{1,\dots,N\},$$

where the j-indexes and k-indexes, respectively, denote the label of the eigenvector and the entry of this eigenvector. The eigenvalues are easily determined by substitution of the eigenvectors into the system $A\underline{v} = \lambda\underline{v}$ and by similar treatment as in 1D. In the current notation, the eigenvector is a two-dimensional array (matrix). In order to transform this into a one dimensional (solution) vector, we use the following transformation

$$\begin{cases} \hat{j} = (j_2 - 1)N + j_1, \\ \hat{k} = (k_2 - 1)N + k_1, \end{cases} \quad \text{for } j_1, j_2, k_1, k_2 \in \{1,\dots,N\}, \quad \text{and} \quad \hat{j},\hat{k} \in \{1,\dots,n\},$$

to make $\underline{v}$ a one-dimensional array of length $n = N^2$. The inverse transformation is given by

$$(j_1, j_2) = \begin{cases} (\mathrm{mod}(\hat{j}, N), \mathrm{trunc}(\hat{j}, N) + 1), & \text{if } \mathrm{mod}(\hat{j}, N) \neq 0, \\ (N, \mathrm{trunc}(\hat{j}, N)), & \text{if } \mathrm{mod}(\hat{j}, N) = 0. \end{cases}$$

Further, we redefine

$$\lambda_{\hat{k}} = \lambda_{k_1,k_2}, \qquad k_1, k_2 \in \{1,\dots,N\}, \qquad \hat{k} \in \{1,\dots,n\}.$$

To solve the equation $A\underline{x} = \underline{b}$, we get

$$\underline{x} = \sum_{j=1}^{n} \frac{1}{\lambda_j}(\underline{b}, \underline{v}_j)\underline{v}_j. \tag{29}$$

For the sake of illustration, we apply the method to the simple Laplace problem

$$\begin{cases} -\Delta u = 1, \ \text{in } \Omega = (0,1)^2, \\ u|_{\partial\Omega} = 0. \end{cases}$$

The solution is compared to the classical use of a direct solver in Matlab, and the solution by both methods is plotted in Fig. 2. No difference can be observed, and the numerical difference was of the order of machine precision. The same was repeated in Fig. 3, where we solved the following boundary value problem:

$$\begin{cases} \Delta^2 u - \Delta u + u = 1, \ \text{in } \Omega = (0,1)^2, \\ u|_{\partial\Omega} = 0, \ \frac{\partial u}{\partial n}|_{\partial\Omega} = 0. \end{cases}$$

No difference between the two approaches has been observed with a difference in the order of machine precision.

For the sake of illustration of the use of boundary conditions, we consider the case

$$\begin{cases} -\Delta u = 1, \qquad \text{in } \Omega = (0,1)^2, \\ u(0,y) = u(x,0) = 0, \ \text{and } \frac{\partial u}{\partial n} = 0, \ \text{for } x = 1 \text{ and } y = 1. \end{cases}$$

The eigenvalues and normalized eigenfunctions for the continuous problem are given by

$$\hat{\lambda}_{k_1,k_2} = \left(\frac{\pi}{2}\right)^2((2k_1 - 1)^2 + (2k_2 - 1)^2), \quad k_1, k_2 = 1, 2, 3, \dots,$$
$$\phi_{k_1,k_2}(x,y) = 2\ \sin\left(\frac{\pi}{2}(2k_1 - 1)x\right)\ \sin\left(\frac{\pi}{2}(2k_2 - 1)x\right), \quad k_1, k_2 = 1, 2, 3, \dots.$$

Using nodes $x_j = jh$, $y_j = jh$, $x_n = y_n = 1$, gives $h = \frac{1}{N}$. This gives the following eigenvalues of the discretization matrix

$$\lambda_{k_1,k_2} = \frac{4}{h^2}\left(\sin^2(\frac{\pi}{4}(2k_1 - 1)) + \sin^2(\frac{\pi}{4}(2k_2 - 1))\right), \qquad k_1, k_2 \in \{1,\dots,N\}.$$

Similar procedures as in the case of fully Dirichlet conditions can be used here to get the inverse of the discretization matrix and the solution to a matrix equation $A\underline{x} = \underline{b}$.

4.3. Higher dimensional Laplace matrix on a (hyper) cube

For the time being, we consider a unit (hyper)cube domain in $\mathbb{R}^d$ with a Laplacian under Dirichlet boundary conditions, which reads as

$$\begin{cases} -\Delta u = f(\mathbf{x}), \ \text{for } \mathbf{x} \in \Omega = (0,1)^d, \\ u|_{\partial\Omega} = 0. \end{cases}$$

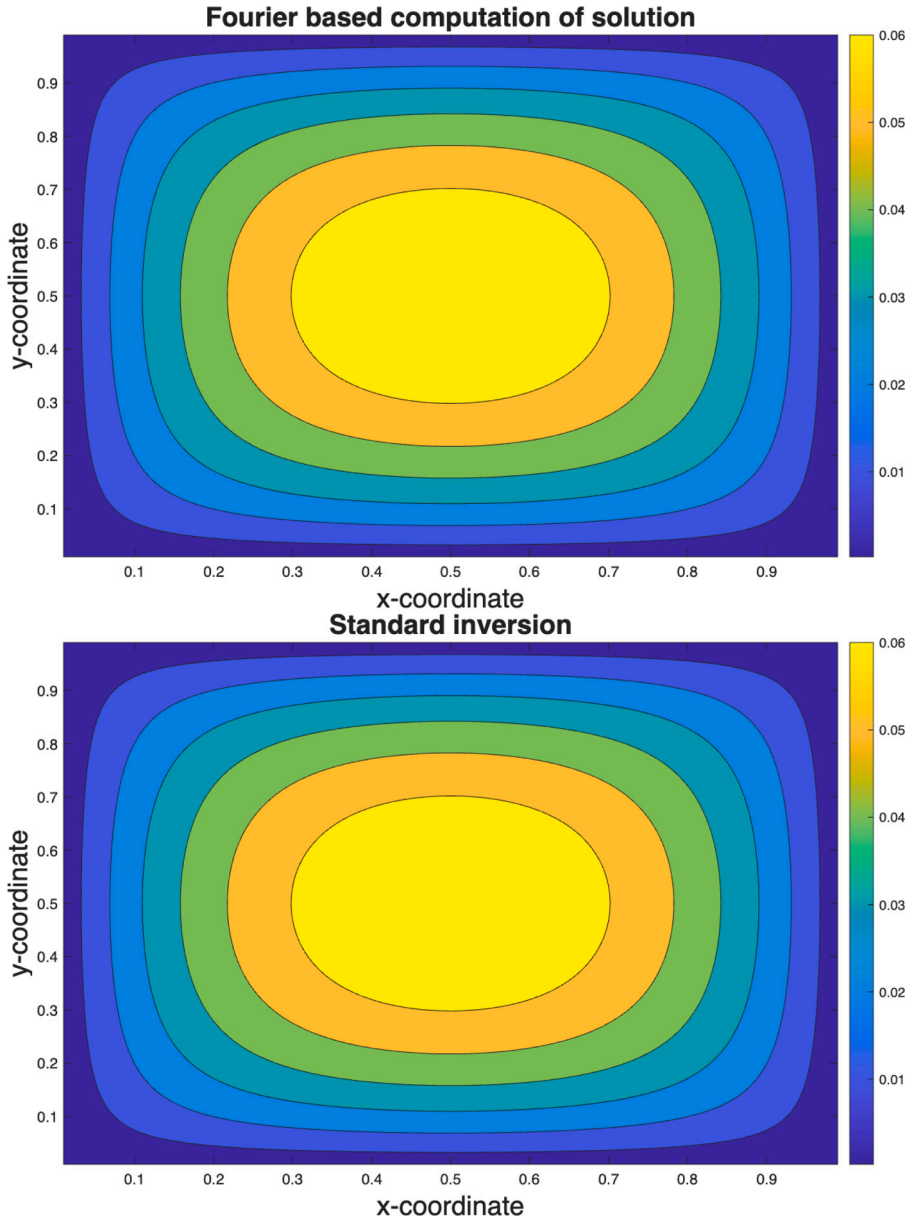


Fig. 2. Solutions to the Poisson equation $-\Delta u = 1$ with $u|_{\partial\Omega} = 0$. The top figure represents the solution obtained by the current procedure, the bottom figure represents the solution by classical solution methods.

The same analysis can be done for a 'hyperbeam' in $\mathbb{R}^d$ as well. The corresponding eigenvalue problem is given by

$$-\Delta\hat{\phi}_{j_1,\ldots,j_d} = \hat{\mu}^2_{j_1,\ldots,j_d}\hat{\phi}_{j_1,\ldots,j_p} = \sum_{p=1}^{d} \mu^2_{j_p}\hat{\phi}_{j_1,\ldots,j_p},$$

where $\mu^2_{j_p}$ represents the j_pth eigenvalue of the operator $-\frac{\partial^2}{\partial x_p^2}(.)$ in the pth coordinate direction. The eigenfunctions $\hat{\phi}_{j_1,\ldots,j_d} : \mathbb{R}^d \longrightarrow \mathbb{R}$, are given by

$$\hat{\phi}_{j_1,\ldots,j_d}(\mathbf{x}) = \prod_{p=\{1,\ldots,d\}} \phi_{p,j_p}(x_p) = 2^{d/2} \prod_{p\in\{1,\ldots,d\}} \sin(\pi j_p x_p), \text{where} \phi_{p,j_p}(x_p) = \sqrt{2}\sin(\pi j_p x_p),$$

for a Dirichlet problem in a hypercube $[0,1]^d$. The eigenvalues are given by

$$\hat{\lambda}_{k_1,\ldots,k_d} = \hat{\mu}^2_{k_1,\ldots,k_d} = \pi^2(k_1^2 + \cdots + k_d^2), \qquad k_j \in \mathbb{N}^{\times}.$$

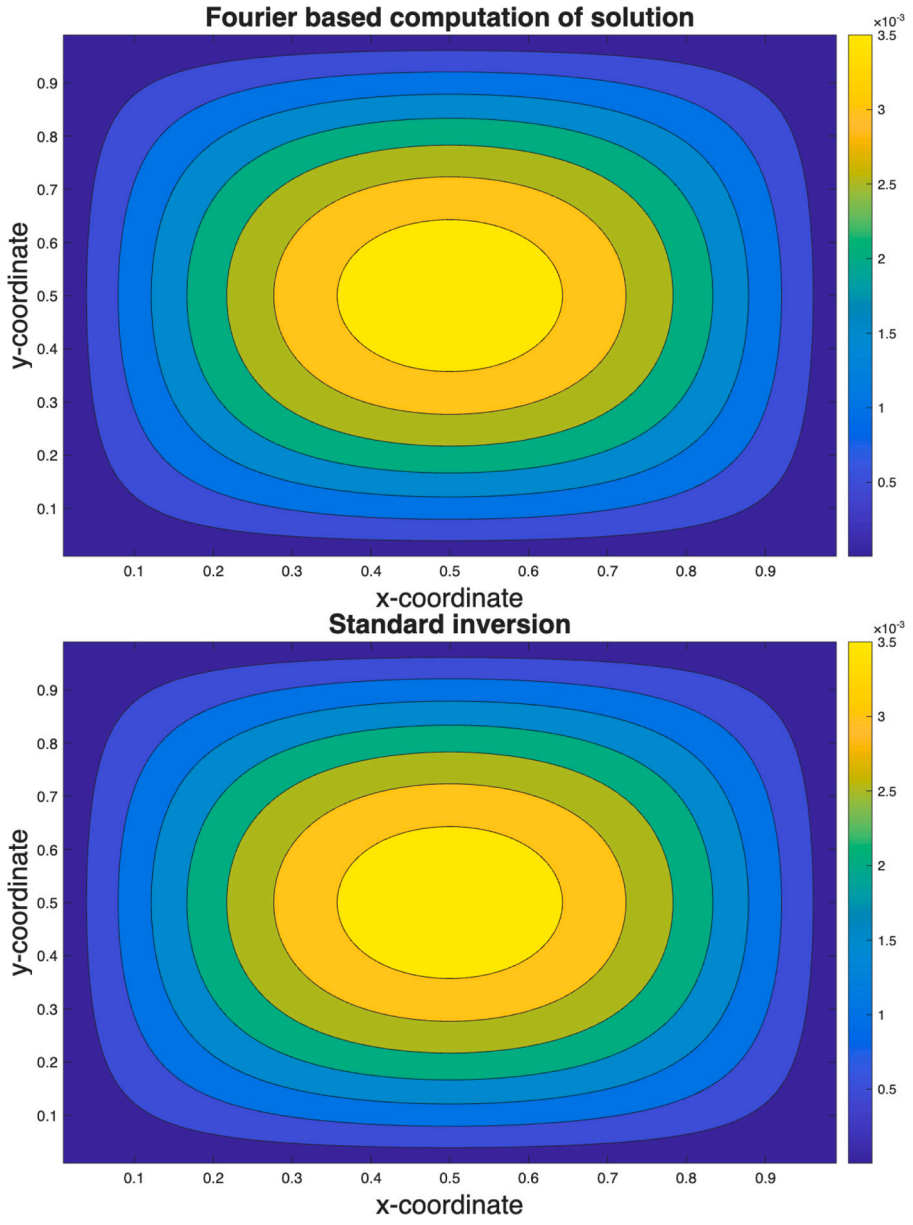


Fig. 3. Solutions to the fourth-order equation $\Delta^2 u - \Delta u + u = 1$ with $u|_{\partial\Omega} = 0$ and $\frac{\partial u}{\partial n}|_{\partial\Omega} = 0$. The top figure represents the solution obtained by the current procedure, the bottom figure represents the solution by classical solution methods.

The multi-dimensional finite difference approach allows the construction of eigenvectors by projecting the eigenfunctions on the meshpoints. Having N unknowns per coordinate direction, this amounts to $n = N^d$ unknowns in a d-dimensional hyperbeam. This gives the following eigenvalues

$$\lambda_{k_1,\ldots,k_d} = \frac{4}{h^2} \sum_{p\in\{1,\ldots,d\}} \sin^2\left(\frac{k_p \pi h}{2}\right), \qquad \text{for } k_j \in \{1,\ldots,N\}, \tag{30}$$

with eigenvectors

$$v_{\hat{j},\hat{k}} = w_{(j_1,\ldots,j_d),(k_1,\ldots,k_d)} = 2^{d/2} \prod_{p\in\{1,\ldots,d\}} \sin(j_p \pi k_p h), \qquad \text{for } j_m,\ k_m \in \{1,\ldots,N\},$$

where the j-indexes and k-indexes, respectively, denote the label of the eigenvector and the entry of this eigenvector. In the current notation, the eigenvector is a two-dimensional array (matrix). In order to transform this into a one-dimensional (solution) vector,

we use the following transformation

$$\begin{cases} \hat{j} = (j_d - 1)N^{d-1} + \cdots + (j_2 - 1)N + j_1 = \sum_{p=2}^{d}(j_p - 1)N^{p-1} + j_1 = \sum_{p=1}^{d}(j_p \cdot N^{p-1}) - \sum_{p=1}^{d-1} N^p, \\ \hat{k} = (k_d - 1)N^{d-1} + \cdots + (k_2 - 1)N + k_1 = \sum_{p=2}^{d}(k_p - 1)N^{p-1} + k_1 = \sum_{p=1}^{d}(k_p \cdot N^{p-1}) - \sum_{p=1}^{d-1} N^p, \end{cases}$$

where we used the simplification that in the discretization we have n unknowns per dimension, hence in $\mathbb{R}^d$, we have n^d unknowns.

4.4. Time dependent problems

The procedure can be used to any first order time-dependent problem. However, for the sake of presentation, we consider a time-dependent diffusion problem for $u = u(\mathbf{x}, t)$, given by

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = 0, \ t > 0, \ \mathbf{x} \in \Omega, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), \ \mathbf{x} \in \Omega, \\ u|_{\partial\Omega} = 0, \ t > 0. \end{cases}$$

After applying a spatial discretization method (semi-discretization), one arrives at

$$\begin{cases} \underline{u}' + A\underline{u} = \underline{0}, \ t > 0, \\ \underline{u}_i(0) = u_0(\mathbf{x}_i). \end{cases}$$

Here $\mathbf{x}_i$ represents the position of the ith nodal point in the spatial discretization. For the sake of presentation, we consider the Euler backward method for the time-integration with time-step Δt. This gives

$$(I + \Delta t A)\underline{u}^\tau = \underline{u}^{\tau-1}.$$

Here τ represents the time-step and $\underline{u}^\tau$ denotes the approximation of $\underline{u}(\tau\Delta t)$. Recursively, this becomes

$$(I + \Delta t A)^\tau \underline{u}^\tau = \underline{u}^0,$$

where $\underline{u}^0 = \underline{u}(0)$. Hence we need to invert the matrix $(I+\Delta t A)^\tau$. Note that this expression can be rewritten in the following polynomial form

$$(I + \Delta t A)^\tau = \sum_{i=0}^{\tau} \binom{\tau}{i} \Delta t^i A^i.$$

It is easy to show that the eigenvectors of A and $(I+\Delta t A)^\tau$ are the same and that the eigenvalues of $(I+\Delta t A)^\tau$ are given by $(1+\Delta t\lambda)^n$, where λ is any eigenvalue of A. This can be written as the following closed-form expression

$$(I + \Delta t \lambda)^\tau = \sum_{i=0}^{\tau} \binom{\tau}{i} \Delta t^i \lambda^j.$$

Hence, for the solution $\underline{u}^\tau$, we get the following formal closed-form expression for the inverse

$$\underline{u}^\tau = \sum_{j=1}^{n} \frac{1}{(1 + \Delta t\lambda_j)^\tau} (\underline{u}_0, \underline{v}_j)\underline{v}_j. \tag{31}$$

Hence, for the inverse of $(I + \Delta t A)^\tau$, we get

$$(((I + \Delta t A)^\tau)^{-1})_{ik} = \sum_{j=1}^{n} \frac{v_{jk}}{n\left(1 + \Delta t\lambda_j\right)^\tau} \, v_{ji}. \tag{32}$$

We note that this can be done similarly for other time-integration methods like the Trapezoidal (Crank–Nicolson) Rule. For the Trapezoidal time-integration method, one arrives at

$$\underline{u}^\tau = \sum_{j=1}^{n} \left(\frac{1 - \frac{\Delta t\lambda_j}{2}}{1 + \frac{\Delta t\lambda_j}{2}} \right)^\tau (\underline{u}_0, \underline{v}_j)\underline{v}_j. \tag{33}$$

This theory can be used to obtain closed-form expressions for the numerical solution to some time-dependent problems involving (polynomials) of discrete Laplace matrices. In Fig. 4, we show some snapshots for a one-dimensional heat equation $u_t = u_{xx}$ with homogeneous Dirichlet boundary conditions and $u = 1$ at $t = 0$. The time-step was $\Delta t = 0.001$, and a backward Euler time-integration method was used. It can be seen that the solution behaves as expected: convergence to zero due to the boundary conditions. A major advantage of the method is that the numerical solution at any time can be obtained from the initial condition, without the need of computing solutions at previous time-steps. The only operation that differs for different time-steps is the power that one has to raise for the amplification factor containing the eigenvalues of the discretization matrix. This makes the method very efficient. We

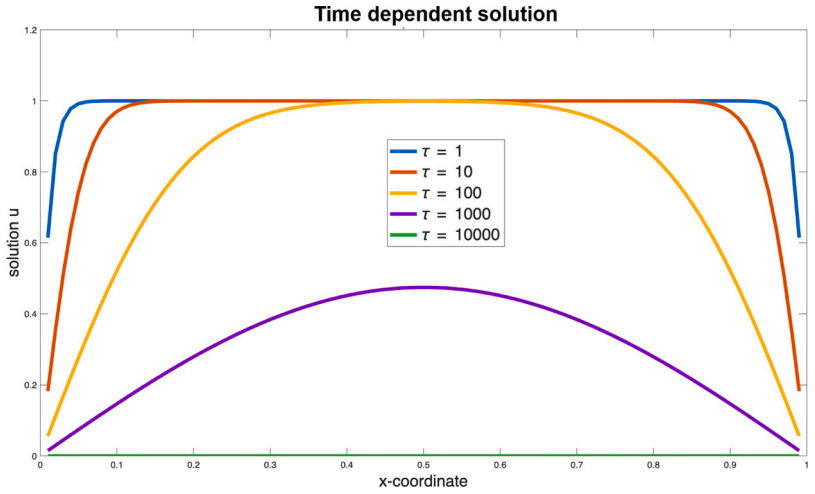


Fig. 4. Snapshot at consecutive times of the heat equation $u_t - u_{xx} = 0$ with $u|_{\partial\Omega} = 0$ and $u = 1$ at $t = 0$ using the backward Euler time-integration method for different τ-values; $\tau = 1, 10, 100, 1000, 10000$.

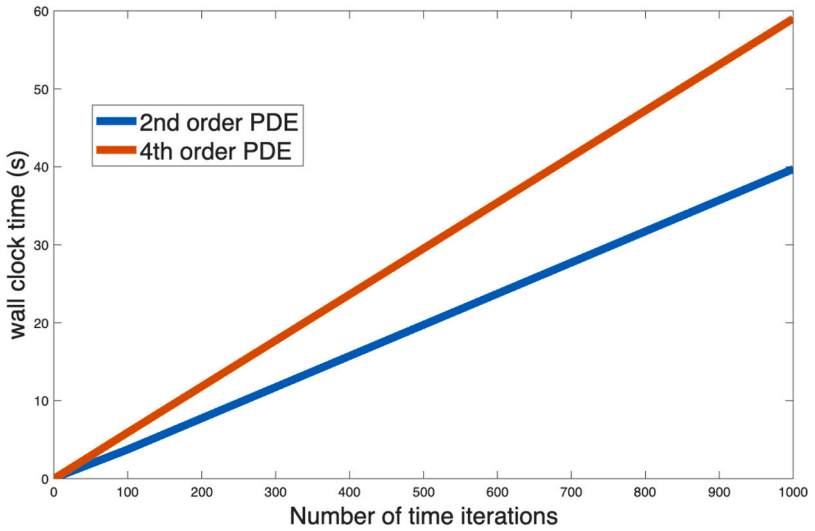


Fig. 5. Computation times (wall clock times) using tic-toc in Matlab for a second and fourth order PDE.

have done simulations with the eigenvalue expansion method and with the ordinary finite difference method. We solved a 2D heat equation (second order PDE), given by

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = 0, \\ u(\mathbf{x}, 0) = 1, \text{ on } \Omega, \\ u|_{\partial\Omega} = 0, \text{ for } t > 0, \end{cases}$$

and a 4th order equation in 2D, given by

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u + \Delta^2 u = 0, \\ u(\mathbf{x}, 0) = 1, \text{ on } \Omega, \\ u|_{\partial\Omega} = \frac{\partial u}{\partial n}|_{\partial\Omega} = 0, \text{ for } t > 0. \end{cases}$$

The computation time (wall clock time) was 0.29 sec (s) in all cases, regardless of the order of the problem and the number of time-iterations, whereas the wall clock time for the finite difference method was determined by the order of the PDE and linearly increases with the number of time-iterations, as expected (see Fig. 5). This shows that the new method is very fast, in particular for large numbers of time-iterations.

5. Discussion and conclusions

We have designed a procedure to invert matrices that are in the polynomial space of Laplace matrices for higher dimensional problems, for all cases where the matrix polynomial is invertible. The procedure can be used to write a closed-form solution to a class of large systems of linear algebraic equations. The method is applicable to problems in (hyper) beams with a regular mesh distribution on linear operators that are based on matrix polynomials of Laplace equations. The method is based on the use of eigenvectors and eigenvalues of the Laplacian, which can be determined exactly. We are aware of the issues with Robin boundary conditions, where the determination of eigenvalues amounts to solving transcendental equations. Then the determination of the inverse requires the (numerical) solution of these transcendental equations, which gives a numerical trait. We further acknowledge that the method is elaborate and that in future studies it will be important to carry out a more detailed comparison of the efficiency of our method with classical iterative schemes such as the (preconditioned) conjugate gradient method or other iterative solution procedures. We are also aware that the problems that we can tackle with this method are idealized in the sense of simple geometries and Laplace-based problems with constant coefficients. Nevertheless, it is possible to tackle polynomials of Laplace matrices, and therefore we think that our approach certainly has some theoretical value. The current method is also helpful to construct closed-form expressions for higher order partial differential equations. Furthermore, time-dependent heat equations with implicit time integration on (hyper) beams have also been treated with the current method since the matrix to be inverted amounts to a polynomial of the discrete Laplace matrix. In case of a backward Euler time integration, a linear relation like $I + \Delta t A$, where A represents the discrete Laplacian, needs to be inverted. For this class of problems, we observed that the computation time does not depend significantly on the number of time-steps and the order of the PDE. Whereas for the classical solving in finite difference methods, the computation time increases linearly with the number of time-iterations. In addition, the computation time also depends on the order of the PDE in case of classical solving in finite difference methods. A limiting factor is that the current method is applicable to linear problems in (hyper) beams. For these problems one can also develop closed-form expressions using separation of variables in continuous problems. However, extension to more complex geometries could be done using eigenvalue and eigenvector determination, where one only includes the most pivotal eigenvalues for time integration. This matter can be investigated in future studies.

Furthermore, since the application to polynomials of matrices is so straightforward, we can use the method to train a neural network with fundamental solutions (in the continuous sense with Dirac delta distributions) for powers of Laplace matrices for different spatial dimensionalities, so that the neural network can provide relatively inaccurate, but very quick solutions to real systems of linear equations with these matrix polynomials. Here, a DeepONet architecture [17] could be an interesting candidate. Further, several recent applications in electronic circuit engineering could use our methodology (see, for instance, [18–20])

Another application of the current method could reside in Laplace filtering, where one uses the (multidimensional) Laplace kernel to capture sharp edges and sharp transitions in the data. It detects edges by using a second-order derivative to measure the rate of change in an image. Often these derivative filters are applied to a smoothed function to avoid problems with image noise amplification [21]. Our method naturally extends to higher-dimensional manifolds (non-curved), making it particularly advantageous for applications in medical imaging, such as processing 3D MRI and CT data, and in scientific visualization, where robust multi-dimensional edge detection is essential.

We finally remark that the current approach is applicable to cases of diagonalizable matrices with orthonormal eigenvectors, which is the case for Hermitian or real, symmetric matrices. However, one must bear in mind that actual practical benefit in terms of explicit solutions, can only be accomplished if the eigenvectors and eigenvalues can be determined explicitly. This is the case for the many discrete Laplace matrices that we consider. The generalization to more classes of matrices certainly deserves more careful attention.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Research England under the Expanding Excellence in England (E3) funding stream, which was awarded to MARS: Mathematics for AI in Real-world Systems in the School of Mathematical Sciences at Lancaster University. Further, we are grateful for the financial support from the Higher Education Commission (HEC) of Pakistan in the framework of project: 1(2)/HRD/OSS-III/BATCH-3/2022/HEC/527 (awarded to S. Asghar).

Data availability

No data was used for the research described in the article.

References

[1] Yousef S. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics; 2003, https://epubs.siam.org/doi/10.1137/1.9780898718003.

[2] Strang G. Computational science and engineering. Wellesley, MA: Wellesley-Cambridge Press; 2007.

[3] Leon D. A wavelet multigrid method using symmetric biorthogonal wavelets. Am J Comput Math 2013;3:127–36. http://dx.doi.org/10.4236/ajcm.2013.32021.

[4] Brigo D, Hanzon B. On some filtering problems arising in mathematical finance. Math Econ 1998;22(1):53–64. http://dx.doi.org/10.1016/S0167-6687(98)00008-0.

[5] Regulwar G, Mahalle A, Pawar R, Shamkuwar S, Kakde P, Tiwari S. Big data collection, filtering, and extraction of features. In: Big data analytics techniques for market intelligence. 2023, p. 136–58. http://dx.doi.org/10.4018/979-8-3693-0413-6.ch005.

[6] Gurarslan G, Karahan H, Alkaya D, Sari M, Yasar M. Numerical solution of advection-diffusion equation using a sixth-order compact finite difference method. Math Probl Eng 2013;672936.

[7] Jiang X, Zhang G, Zheng Y, Jiang Z. Explicit potential function and fast algorithm for computing potentials in $\alpha \times \beta$ conic surface resistor network. Expert Syst Appl 2024;238(Part E):122157. http://dx.doi.org/10.1016/j.eswa.2023.122157.

[8] Jiang X, Meng D, Zheng Y, Jiang Z. A dynamic cobweb resistance network solution based on a structured zeroing neural network and its applications. Sci Rep 2025;15:5222. http://dx.doi.org/10.1038/s41598-025-89471-6.

[9] Crank J, Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. Proc Camb Philos Soc 1947;43:50–67. http://dx.doi.org/10.1017/S0305004100023197.

[10] Mittal RC, Bhatia R. Numerical solution of second order one dimensional hyperbolic telegraph equation by cubic B-spline collocation method. Appl Math Comput 2013;220:496–506. http://dx.doi.org/10.1016/j.amc.2013.05.081.

[11] Strang G. Introduction to linear algebra. Wellesley, MA: Wellesley-Cambridge Press; 2009.

[12] Gantmacher FR. In: The theory of matrices, vol. 1, New York: AMS Chelsea Publishing; 1960.

[13] Roman S. Advanced linear algebra. New York, NY: Springer New York; 1992.

[14] Fine B, Rosenberger G. The fundamental theorem of algebra. Springer New York; 1997.

[15] Vermolen FJ, Bakker DR, Vuik C. On the fundamental solutions-based inversion of Laplace matrices. Results Appl Math 2022;15:100288. http://dx.doi.org/10.1016/j.rinam.2022.100288.

[16] Gueye SB. Exact inversion of pentadiagonal matrix for semi-analytic solution of 2D poisson equation. J Mod Phys 2022;13(12). http://dx.doi.org/10.4236/jmp.2022.1312094.

[17] Li H, Miao Y, Khodaei ZS, Aliabadi MH. An architectural analysis of DeepOnet and a general extension of the physics-informed DeepOnet model on solving nonlinear parametric partial differential equations. Neurocomputing 2025;611. http://dx.doi.org/10.1016/j.neucom.2024.128675.

[18] Jiang X, Guan Y, Zheng Y, Jiang Z, Liu Z. Robot path planning based on potential function of a resistor network. Appl Math Model 2026;151(Part B):116534. http://dx.doi.org/10.1016/j.apm.2025.116534.

[19] Jiang Z, Zhou Y, Jiang X, Zheng Y. Analytical potential formulae and fast algorithm for a horn torus resistor network. Phys Rev E 2023;107:044123. http://dx.doi.org/10.1103/PhysRevE.107.044123.

[20] Jiang X, Dai J, Zheng Y, Jiang Z. Modeling of omnidirectional antenna space radiation shaped resistor networks and efficient path planning. Appl Phys Rev 2025;12(4):041405, 10.1063/5.0291210.

[21] Gonzalez RC, Woods RE. Digital image processing. second ed. Prentice-Hall; 2002.