Journal of Peridynamics and Nonlocal Modeling (2021) 3:307–326
https://doi.org/10.1007/s42102-021-00053-2

ORIGINAL ARTICLES

Connections Between the Meshfree Peridynamics Discretization and Graph Laplacian for Transient Diffusion Problems

Longzhen Wang[1] · Florin Bobaru[1]

Received: 17 December 2020 / Accepted: 1 March 2021 / Published online: 24 March 2021

Abstract
Formulations for diffusion processes based on graph Laplacian kernels have been recently used to solve linear transient heat transfer problems with insulated boundary conditions by way of a spectral-based semi-analytical approach. This has been called the "spectral graph" (SG) approach. In this paper, we show that the meshfree discretization for corresponding peridynamic (PD) models leads to a graph structure that allows us to introduce the "graph Laplacian PD" (GL-PD) method using the semi-analytical approach employed in the SG approach to solve the transient heat diffusion problems. The new method, GL-PD, can be seen as a "hybrid" between SG and PD with meshfree discretization. We discuss the similarities and differences between the GL-PD and the SG approaches. Main differences are related to calibration and discretization procedures. We use a 1D heat diffusion example to highlight some limitations the spectral-based semi-analytical method in the SG approach has compared with the direct time-integration normally used in computing solutions to transient diffusion problems. We then propose an extension of the semi-analytical approach to solve transient diffusion problems with Dirichlet boundary conditions, using a recent scaling and squaring algorithm that calculates the matrix exponential of non-symmetric matrices.

Keywords Peridynamics · Graph Laplacian · Spectral graph theory · Diffusion · Matrix exponential

1 Introduction to Graph Laplacian Exponential Diffusion Kernel and Peridynamics

Graphs are data structures widely used in many applications [1]. Graphs can be used to model complex structures, for instance, molecules [2], proteins [3], social networks [4], images [5], Internet [6], and natural languages [7]. In chemistry and biology, identifying the

✉ Florin Bobaru
fbobaru2@unl.edu

[1] Department of Mechanical and Materials Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0526, USA

function of a molecule with experimental methods can be expensive and time-consuming. Alternatively, one can use graph theory and machine learning [8] to estimate properties of the new molecule/protein based on other molecules/proteins with similar structures.

By using machine learning methods to determine which class a graph belongs to, one needs to quantify the similarity between graphs. However, due to their complex structure, comparing graphs is an NP-problem (nondeterministic polynomial time) [1]. To overcome the unaffordable complexity, kernel-based methods that can capture the entire structure of a graph have been used [1]. One advantage of kernel-based methods is that data is implicitly mapped into a Hilbert space and can be easily compared [9].

In this paper, we will focus on the graph Laplacian exponential diffusion kernel, which is the solution to the diffusion equation on the graph domain, obtained with a spectral-based semi-analytical approach [9, 10]. The graph Laplacian exponential diffusion kernel was first introduced in 2002 and has been used in different applications, including image classification tasks [9], community detection in networks [11], etc. In recent papers, the graph Laplacian exponential diffusion kernel was used to solve heat diffusion in the additive manufacturing process. The additive manufactured parts are modeled as graphs and the classical heat diffusion partial diffusion equation (PDE) is solved [12, 13]. In these works, the procedure was called "spectral graph (SG) approach". Notably, until now, as remarked in [13], the SG approach has been limited to problems with insulated boundaries only, i.e., zero Neumann boundary conditions. By constructing the graph and linking each node with all other nodes within a certain distance [13], effectively a nonlocal model is created. As we shall see, at least formally, the SG approach is similar to a peridynamic model for diffusion, while some differences related to calibration and discretization procedures exist between them.

Peridynamics was introduced as a nonlocal form of continuum mechanics in [14]. Since then, it has been successfully applied to modeling of brittle fracture [15–20], ductile fracture [21–23], fracture and damage in composites [24–27], heat transfer [28–32], corrosion damage [33–36], etc. In peridynamics, each material point is connected through bonds to other points, within a certain neighborhood region called 'the horizon'. The properties of these bonds are obtained by, for example, matching the classical strain energy density under a homogeneous deformation (for elasticity) [37, 38] or heat flux under constant thermal gradient conditions [28]. Various methods can be used to discretize peridynamic formulations. The so-called "meshfree discretization" uses a one-point Gaussian quadrature scheme for approximating the integral operator in the formulation [38]. This numerical method is popular for peridynamic models because of its flexibility in representing unrestricted damage evolution, which helps deliver on one of the main advantages peridynamic models have compared with classical, PDEs-based models. With the meshfree spatial discretization, the peridynamic model can be treated as a graph, in which vertices are the nodes and edges are the peridynamic bonds.

The paper is organized as follows: graph concepts and graph Laplacian exponential diffusion kernel ("spectral graph approach") are reviewed in Sect. 2.1. The peridynamic diffusion model for transient diffusion is reviewed in Sect. 2.2. In Sect. 3, using the 2D transient heat transfer setup, we explain the similarities and differences between the GL-PD approach and the SG approach. Differences related to the calibration and discretization procedures are highlighted. In Sect. 4, a numerical example for 1D transient heat transfer is used to show the limitations of the spectral-based semi-analytical approach. In Sect. 5, we introduce an extension of the semi-analytical approach to 1D and 2D diffusion problems with Dirichlet boundary conditions. In Sect. 6, we present conclusions and future work.

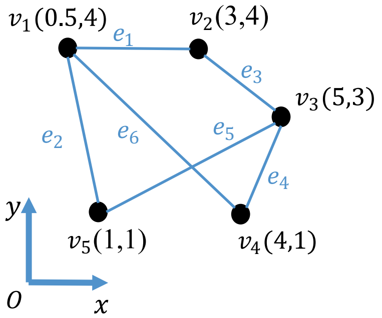
2 The Graph Laplacian Exponential Diffusion Kernel and the Peridynamic Model for Diffusion

2.1 Review of Graphs and the Graph Laplacian Exponential Diffusion Kernel

In mathematics, a graph G consists of a set of vertices $V = \{v_1, v_2, \dots\}$ and another set of edges $E = \{e_1, e_2, \dots\}$ such that each edge is identified with an unordered pair of vertices [39]. Graphs are commonly represented as diagrams in which vertices are nodes and edges are represented by lines connecting nodes [39], as shown in Fig. 1. In this paper, we study the weighted undirected simple graphs (no multiple edges or loops).

Although graphs can be represented as diagrams, matrices are a convenient way to represent graphs, especially in computational studies [39]. There are different matrices used in graph theory [39]; here, we focus only on the adjacency and Laplacian matrices.

The adjacency matrix of an N-vertex weighted undirected simple graph G is an N by N real-symmetric matrix $A(G) = [a_{ij}]$, with a_{ij} being the weight of the edge between nodes i and j, if there is an edge between the i^{th} and j^{th} vertices, and $a_{ij} = 0$ if not. In the weighted undirected graph in Fig. 1, the vertices' coordinates in a two-dimensional Cartesian coordinate system are shown, and the weights of edges are equal to the inverse of the edges length square. For example, the length of edge between vertex v_1 and vertex v_2 is 2.5 and the



$$A = \begin{bmatrix} 0 & 0.16 & 0 & 0.0471 & 0.1081 \\ 0.16 & 0 & 0.2 & 0 & 0 \\ 0 & 0.2 & 0 & 0.2 & 0.05 \\ 0.0471 & 0 & 0.2 & 0 & 0 \\ 0.1081 & 0 & 0.05 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.3152 & 0 & 0 & 0 & 0 \\ 0 & 0.36 & 0 & 0 & 0 \\ 0 & 0 & 0.45 & 0 & 0 \\ 0 & 0 & 0 & 0.2471 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.3152 & -0.16 & 0 & -0.0471 & -0.1081 \\ -0.16 & 0.36 & -0.2 & 0 & 0 \\ 0 & -0.2 & 0.45 & -0.2 & -0.05 \\ -0.0471 & 0 & -0.2 & 0.2471 & 0 \end{bmatrix}$$

Fig. 1 Undirected graph (v: vertices, e: edges) and its corresponding adjacency, degree, and Laplacian matrices.

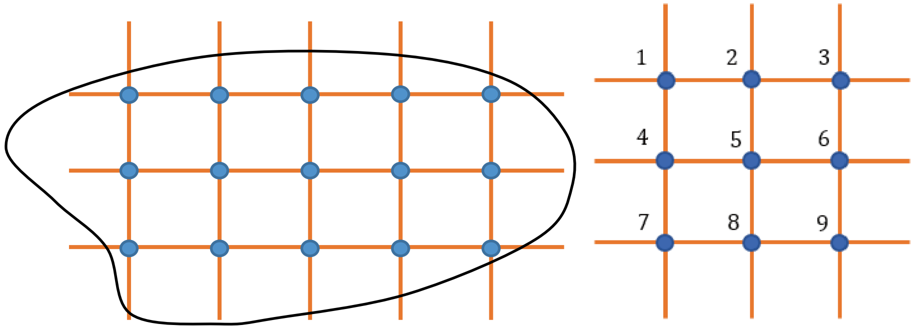


Fig. 2 A domain discretized by a uniform grid (left) and the 5-point stencil for finite difference approximations (right)

weight of this edge is 0.16. The degree matrix of an N-vertex weighted undirected simple graph G is an N by N diagonal matrix $D(G) = \left[d_{ij}\right]$, with d_{ii} being *the degree* of the i^{th} vertex, which is the sum of the weights of edges connected to the i^{th} vertex.

The Laplacian matrix of an N-vertices weighted undirected simple graph G is defined as $L(G) = D(G) - A(G)$. The matrix representation of the graph is shown in Fig. 1.

The negative Laplacian matrix can be seen as a proper discretization of the classical Laplacian operator [40, 41]. For example, on a square grid in 2D (see Fig. 2) with grid spacing $\Delta x = \Delta y = h$, the finite difference approximation of the continuous Laplace operator at node 5, of coordinates (x_5, y_5), is $\frac{\partial^2\theta}{\partial x^2}(x_5, y_5) + \frac{\partial^2\theta}{\partial y^2}\left(x_5, y_5\right) \approx \frac{\theta_4 - 2\theta_5 + \theta_6}{(\Delta x)^2} + \frac{\theta_2 - 2\theta_5 + \theta_8}{(\Delta y)^2} = \frac{1}{h^2}(\theta_4 + \theta_6 + \theta_2 + \theta_8 - 4\theta_5)$. If the square grid is seen as a graph, node 5 is connected to four neighboring nodes (nodes 2, 4, 6, and 8). In the graph, the edge weight is the inverse of edge length square, same as in Fig. 1. The continuous Laplacian at node 5 can then be approximated by the negative Laplacian matrix multiplied by vector $\boldsymbol{\theta}$, which leads to: $\frac{\partial^2\theta}{\partial x^2}(x_5, y_5) + \frac{\partial^2\theta}{\partial y^2}\left(x_5, y_5\right) \approx \frac{1}{h^2}(\theta_4 + \theta_6 + \theta_2 + \theta_8 - 4\theta_5)$, (see Fig. 3). Thus, $-\boldsymbol{L}$ is the same as the finite difference approximation of the continuous Laplacian operator [9]. In the limit $h \to 0$, this approximation becomes exact [9].

In some recent publications [12, 13], the spectral graph theory (which studies eigenvectors of matrices of graphs, e.g., adjacent matrix and Laplacian matrix) is used to predict heat diffusion in parts built by additive manufacturing processes. In these references, the problem domain is discretized first into nodes, then each node is connected to all the other nodes

$$\nabla \begin{bmatrix} \vdots \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \vdots \end{bmatrix} = {}^{1}\!/_{h^2} \begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\ \cdots & 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 & \cdots \\ \cdots & 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \vdots \end{bmatrix}$$

Fig. 3 The approximation of the continuous Laplacian using the Laplacian matrix

which are located within a certain distance b, called “neighborhood distance”. The concept of the neighborhood for a graph vertex in the SG approach is identical to that of the horizon region in PD. A graph can then be defined by treating the nodes in this discretization as its vertices and the connections among nodes as the graph’s edges. Using this graph, one can solve heat transfer problems as discussed in [9] (with a spectral-based semi-analytical approach, see below) and use this for graph classification problems. References [12, 13] used this spectral graph (SG) approach to solve transient heat transfer problems with insulated boundary conditions and applied it to specific additive manufacturing problems.

In this paper, we use a 2D transient heat transfer problem as an example to show the relationship between the SG approach and PD. The problem setup is shown in Fig. 4.

Heat flow in 2D can be described, under some limiting conditions (see Sect. 4) by the following linear partial differential equation (PDE) [42]:

$$\rho c \frac{\partial \theta(\boldsymbol{x}, t)}{\partial t} = \kappa \nabla^2 \theta(\boldsymbol{x}, t) = \kappa \left(\frac{\partial^2 \theta(\boldsymbol{x}, t)}{\partial x^2} + \frac{\partial^2 \theta(\boldsymbol{x}, t)}{\partial y^2} \right) \tag{1}$$

where $\theta(\boldsymbol{x}, t)$ is the temperature (K) at point $\boldsymbol{x}$ and time t, $\boldsymbol{x}$ is the vector of components (x, y), t is time (s), ρ is the density (kg/m^3), c is the specific heat capacity ($J\,kg^{-1}\,K^{-1}$), κ is the thermal conductivity ($W\,m^{-1}\,K^{-1}$), and ∇^2 is the Laplacian operator. For simplicity, here we set the values of ρ, c, κ to equal one and define the following initial-boundary value problem (IBVP):

$$\begin{aligned} \frac{\partial \theta(\boldsymbol{x}, t)}{\partial t} &= \nabla^2 \theta(\boldsymbol{x}, t), \forall \boldsymbol{x} \in \Omega \\ \frac{\partial \theta(\boldsymbol{x}, t)}{\partial n} &= 0, \forall \boldsymbol{x} \in \partial\Omega \\ \theta(\boldsymbol{x}, t = 0) &= \theta_0(\boldsymbol{x}), \forall \boldsymbol{x} \in \Omega \end{aligned} \tag{2}$$

Following [12, 13], to numerically solve this linear IBVP, the problem domain is discretized into nodes, not necessarily equally spaced, each node being connected to all neighboring nodes which are located within a certain distance, the neighborhood distance b. In the 1D example in [13], the weight of edge was defined as $A_{ij} = 1/c_{ij}^2$, $c_{ij} \leq b$, $i \neq j$, where c_{ij} is the Euclidean distance between node i and node j, and b is the radius defining the neighborhood. For the 3D example in [13], the weight of edge was chosen to be a

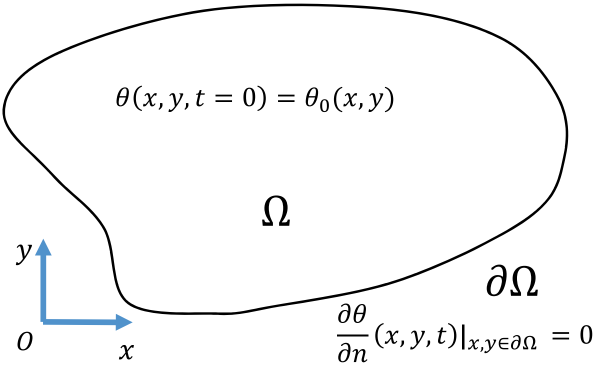

$\theta(x, y, t = 0) = \theta_0(x, y)$
Ω
$\partial\Omega$
y
O
x
$\frac{\partial \theta}{\partial n}(x, y, t)|_{x,y \in \partial\Omega} = 0$

Fig. 4 Domain and boundary conditions for a linear 2D transient heat transfer problem with insulated boundaries

bell-shaped function, $A_{ij} = e^{-c_{ij}^2/\sigma^2}$, $c_{ij} \leq b$, $i \neq j$, where σ is the standard deviation for distances c_{ij}. After the problem domain is discretized and a graph is constructed, the Laplacian operator ∇^2 is replaced by the specific negative Laplacian matrix L, corresponding to that particular discretization. After replacing the continuous Laplacian, the discretized form is called the “heat equation on the graph” [9]. After the spatial discretization, the temperature field $\theta(\boldsymbol{x}, t)$ becomes a vector $\boldsymbol{\theta}(t) = \left[\, \theta_1(t) \;\; \theta_2(t) \;\; \cdots \;\; \theta_{N-1}(t) \;\; \theta_N(t) \,\right]$, where N is the total number of nodes.

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial t} = -L\boldsymbol{\theta}(t) \tag{3}$$

This linear, first-order system of differential equations has an exact solution [43]:

$$\boldsymbol{\theta}(t) = e^{-Lt}\boldsymbol{\theta}_0 \tag{4}$$

where $\boldsymbol{\theta}_0$ is the vector that contains the initial temperature at all nodes.

To calculate the matrix exponential term, eigen-decomposition is used below. The eigenvalues λ of this $N \times N$ Laplacian matrix L are found solving the equation:

$$\det(L - \lambda I) = 0 \tag{5}$$

For each eigenvalue λ, the corresponding eigenvector $\boldsymbol{v}$ is the solution to the linear system:

$$(L - \lambda I)\boldsymbol{v} = 0 \tag{6}$$

Once eigenvalues and eigenvectors are found, the Laplacian matrix L can be written as

$$L = \phi \Lambda \phi^{-1} \tag{7}$$

where ϕ is the eigenvector matrix whose columns are eigenvectors of L and Λ is the diagonal matrix whose diagonal elements are the eigenvalues of L [44]. Based on the definition of the Laplacian matrix $L(G) = D(G) - A(G)$, the adjacency matrix A of an undirected simple graph is symmetric. Since the degree matrix D is diagonal, the Laplacian matrix is symmetric [41], and, therefore, its eigenvectors ϕ are orthogonal [44]. Thus, the Laplacian matrix can be expressed as

$$L = \phi \Lambda \phi^{-1} = \phi \Lambda \phi^T \tag{8}$$

By replacing L in Eq. (4), the exact solution can be written as

$$\boldsymbol{\theta}(t) = e^{-\phi \Lambda \phi^T \mathrm{t}}\boldsymbol{\theta}_0 \tag{9}$$

where the term $e^{-\phi \Lambda \phi^T \mathrm{t}}$ is called the graph Laplacian exponential diffusion kernel [45, 46].

We can further simplify this expression by using the Taylor expansion about time 0, to find:

$$\begin{aligned} e^{-\phi\Lambda\phi^T t} = I - t\frac{\phi\Lambda\phi^T}{1!} + t^2\frac{\left(\phi\Lambda\phi^T\right)^2}{2!} - t^3\frac{\left(\phi\Lambda\phi^T\right)^3}{3!} + t^4\frac{\left(\phi\Lambda\phi^T\right)^4}{4!} - = I - \\ \frac{\phi(\Lambda t)\phi^T}{1!} + \frac{\phi(\Lambda t)^2\phi^T}{2!} - \frac{\phi(\Lambda t)^3\phi^T}{3!} + \frac{\phi(\Lambda t)^4\phi^T}{4!} - = \phi e^{-\Lambda t}\phi^T \end{aligned} \tag{10}$$

With this change, the semi-analytical solution to the diffusion IBVP with insulated boundary conditions becomes

$$\boldsymbol{\theta}(t) = \phi e^{-\Lambda t}\phi^{T}\boldsymbol{\theta}_0 \tag{11}$$

To make a connection between edges' weights in the graph and the diffusion material parameters (density, specific heat capacity, thermal conductivity), a so-called "gain factor" g is used in the exponential factor (see [13]):

$$\boldsymbol{\theta}(t) = \phi e^{-g\Lambda t}\phi^{T}\boldsymbol{\theta}_0 \tag{12}$$

To obtain the "gain factor", a problem-dependent calibration (note the difference between the gain factors obtained in references [12] and [13]) was proposed. The numerical solution in Eq. (12) is compared with exact solutions of the classical heat equation (if they exist), or with numerical solutions obtained by other methods ([13] used the finite element method). From [12] and [13], it appears that this calibration procedure has to be performed anytime a different domain shape is used. Having to find another numerical solution for the calibration step seems to defy the purpose of creating an efficient model, as intended in [13]. Nevertheless, the calibration step presented in [13] is as follows: for a given domain, piecewise constant initial conditions are applied and the problem with insulated boundaries conditions is solved; the temperature at one node (located at the center of the initially heated region) is monitored to steady state and the factor g that minimizes the least-square error (in time) between the solution from Eq. (12) and the exact (or alternative numerical) solution is the calibrated "gain factor". Note that it is also likely that if one changes the location of the point used in the calibration procedure (for instance, using a point on the boundary of the region instead of its "center"), the value of the gain factor would change, even if the domain and discretization are kept the same.

With this calibrated solution, the temperature field at any time can then be directly calculated with the given constructed graph and initial temperature from Eq. (12). Note that this solution method can only be used under certain special conditions (insulated boundary conditions, linear problems, no heat sources or sinks, etc.), and has some additional limitations. All these are discussed in Sect. 4.

2.2 The Peridynamic Model for Diffusion

In this section, we briefly review the bond-based peridynamic (PD) diffusion model.

The bond-based peridynamic model for diffusion problem, studied in [28], replaces the Laplacian term with an integral term of temperature (or concentration) differences. Since spatial integration is well defined irrespective of possible discontinuities, the model can naturally handle problems in which such discontinuities may appear and evolve [15, 29]. This is the reason why peridynamic formulations are widely used to solve problems with discontinuities such as cracks [16], damage [47], dissolution [48], etc.

In peridynamics, each point is connected and interacts with all other points within a certain region, called the horizon region, as shown in Fig. 5. The bond-based PD equation for diffusion is [28]:

$$\rho c\frac{\partial\theta(\boldsymbol{x},t)}{\partial t} = \int_{H_x} k(\boldsymbol{x},\hat{\boldsymbol{x}})\frac{\theta\left(\hat{\boldsymbol{x}},t\right)-\theta(\boldsymbol{x},t)}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|^{2}}d\hat{\boldsymbol{x}} \tag{13}$$

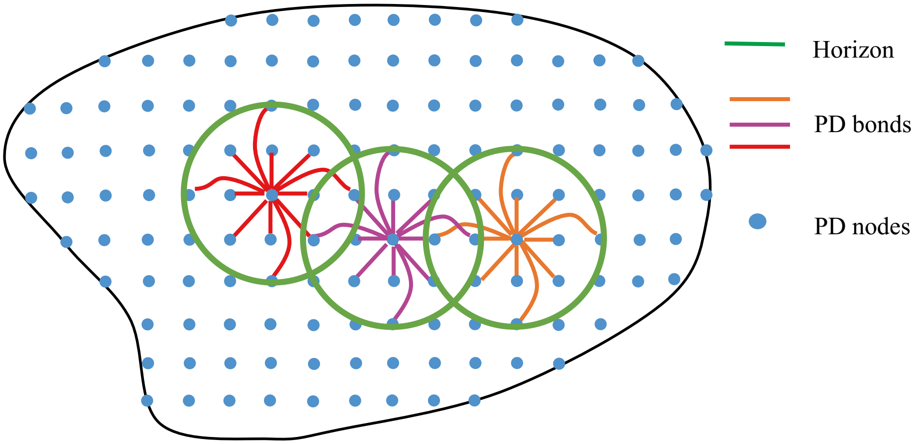


Fig. 5 Schematic diagram of a peridynamic discretization (based on the one-point Gaussian quadrature in space) showing a few nodes and their bonds that create a graph structure for this PD discretization

where $H_{\mathbf{x}}$ is the horizon of $\boldsymbol{x}$, $k(\boldsymbol{x},\widehat{\boldsymbol{x}})$ is the $(\boldsymbol{x},\widehat{\boldsymbol{x}})$ bond micro-conductivity ($W\,m^{-3}\,K^{-1}$). Instead of using $\|\widehat{\boldsymbol{x}}-\boldsymbol{x}\|^{2}$ in the kernel denominator, one can also use, for example, $\|\widehat{\boldsymbol{x}}-\boldsymbol{x}\|^{1}$ or $\|\widehat{\boldsymbol{x}}-\boldsymbol{x}\|^{0}$ (see [49, 50]).

The peridynamic heat flux at point $\boldsymbol{x}$ across surface S is defined by [28]:

$$Q(\boldsymbol{x},S,t)=-\int_{H_{\mathbf{x}}/S}k(\boldsymbol{x},\widehat{\boldsymbol{x}})\frac{\theta\left(\widehat{\boldsymbol{x}},t\right)-\theta(\boldsymbol{x},t)}{(\widehat{\boldsymbol{x}}-\boldsymbol{x})\bullet\boldsymbol{n}}d\widehat{\boldsymbol{x}} \tag{14}$$

where $H_{\mathbf{x}}/S$ is the portion of $H_{\mathbf{x}}$ on one side of the surface S and $\boldsymbol{n}$ is the exterior normal to the surface.

The micro-conductivity function $k\left(\boldsymbol{x},\widehat{\boldsymbol{x}}\right)$ describes the properties of bonds between material points inside the horizon. In PD, the calibration is normally done at the continuum level [28], but it can also be done at the discrete level [51]. In this paper, we use a calibration performed at the continuum level, given by an analytical formula. This means that it does not have to be repeated for each and every discretization tried. To obtain a physically based micro-conductivity function, the peridynamic heat flux is calibrated to the classical heat flux for a given linear temperature profile [29]. With this calibration, and for the particular profile of a micro-conductivity function that is constant over the horizon, in 2D, one obtains $k(\mathbf{x},\widehat{\mathbf{x}})=\frac{4\kappa}{\pi\delta^{2}}$ (see [28, 29]), where κ is the thermal conductivity and δ is the horizon size (the radius of the horizon region when this is taken to be a circular disk). Other profiles for the micro-conductivity function over the horizon region can be selected, including linear ("conical"), Gaussian, semi-elliptical, etc. [52, 50]. Different micro-conductivity functions may lead to different convergence rates to the classical solution (see [52]) as the horizon δ decreases to zero while the ratio between δ and the discretization spacing is maintained constant (the so-called δ-convergence, see [37]).

To numerically approximate solutions to Eq. (13), a variety of numerical methods can be used, including the one-point Gaussian quadrature (which leads to a "meshfree" discretization method) [28, 37], the finite element method [53, 54], the fast convolution-based method [55, 56], etc. In the meshfree method, there are no elements or other geometrical connections between nodes [38]. The advantage of the meshfree method in discretizing PD models rests in its flexibility for representing unrestricted damage evolution. Damage in this method is determined by individual bond failure events. This definition of damage is a much richer quantity than, for example, the scalar variable in continuum damage mechanics (see [57]). In some sense, PD damage is a mapping, not necessarily continuous, from the vector space of bonds at a point $\boldsymbol{x}$ to a vector space of dimension equal to the number of bonds at a node, whose elements have entries zero or one, depending on whether the particular bond is broken or intact. In the discretized version, PD damage is therefore a vector state (see [58]). Note that the "damage index" usually employed in PD models (the ratio between the number of broken bonds to the number of bonds at a node) is simply a way to monitor the actual evolution of the PD damage and allow for an easy representation of it. Other than the meshfree method, the continuous and discontinuous Galerkin finite element methods have also been used in PD modeling [53]. The advantage of using a finite element method is the potential use of established finite element software to solve peridynamic models. However, these methods cannot handle arbitrary evolution of damage and fracture with the ease that the meshfree discretization can ([59, 60]).

3 The Spectral-Based Semi-Analytical Approach for the Bond-Based Peridynamic Diffusion Model

In this section, we show that the meshfree discretization for PD leads to a graph structure; on this graph, we construct the graph Laplacian matrix and solve the transient heat transfer problem using the semi-analytical method as done in the SG method. We call this, for clarity, the "graph Laplacian PD" (GL-PD) method. While the GL-PD is similar to the SG approach, some important differences are:

- the calibration procedures are different: in GL-PD, a general calibration is used for a specific choice of the micro-conductivity functional form; these parameters can be obtained analytically, and then be used for problems with arbitrary domain geometry and boundary conditions; in SG, a calibration has to be performed for each new problem (problems with different material, different node discretization, or different neighborhood radius); and
- the specific definition of the terms in the adjacency matrix is different: the SG and GL-PD could use the same discretization schemes, but, as implemented in [12] and [13], SG uses randomly distributed nodes that do not carry an associated volume, while in GL-PD, nodes come with their associated volume (area in 2D, length in 1D).

With the meshfree discretization in GL-PD, the domain is discretized into cells (with nodes at the "center" of the cell) and each node is connected through bonds to all other nodes within its horizon. To discretize Eq. (13), the integral is replaced by a finite Riemann sum (based on the one-point Gaussian quadrature). After discretization, the bond-based peridynamics model can be seen as an undirected and weighted graph. The nodes can be treated as the vertices in SG. The concept of bonds is similar to that of the graph edges in SG. However, the conductivity properties are different because they are obtained using different calibration procedures: in SG, they depend on the edge length and gain factor, while in GL-PD, they depend on the micro-conductivity and may or may not depend on the bond length (see the constant or varying micro-conductivity over the horizon; this choice is at the latitude of the modeler).

If we use a uniform node discretization, then all nodes have the same area in 2D (volume in 3D and length in 1D), except, at most, for nodes at the boundary [61]. Note that non-uniform discretization can also be used in PD models [62–64]. With the meshfree discretization, Eq. (13) becomes ([28]):

$$\rho c\dot{\theta}\left(\boldsymbol{x}_i,t\right)=\sum\nolimits_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\frac{\theta\left(\boldsymbol{x}_j,t\right)-\theta(\boldsymbol{x}_i,t)}{\left(\boldsymbol{x}_j-\boldsymbol{x}_i\right)^2}V_{ij} \tag{15}$$

where V_{ij} is the portion of the "volume" of node $\boldsymbol{x}_j$ actually covered by the horizon of node $\boldsymbol{x}_i$.

Similar to the choice made in Sect. 2.1, for simplicity, we set the values of ρ, c, κ to be equal to one. We use the constant micro-conductivity function with $k\left(\mathbf{x}_i,\mathbf{x}_j\right)=\frac{4}{\pi\delta^2}$ (2D). The discretized form becomes (notice the switch in the numerator, useful later for matching the definition of the Laplacian matrix):

$$\dot{\theta}\left(\boldsymbol{x}_i,t\right)=\sum\nolimits_j\frac{-4}{\pi\delta^2}\frac{V_{ij}}{\left(\boldsymbol{x}_j-\boldsymbol{x}_i\right)^2}\left[\theta\left(\boldsymbol{x}_i,t\right)-\theta(\boldsymbol{x}_j,t)\right] \tag{16}$$

In the meshfree peridynamic diffusion model, bond properties (length, micro-conductivity) and nodal "volumes" define the weights of the graph edges. For the edge between nodes i and j, the weight is $A_{ij}^{\mathrm{PD}}=\frac{k(x_i,x_j)V_{ij}}{\left(x_j-x_i\right)^2}$. In the case of constant micro-conductivity, the weight becomes $A_{ij}^{\mathrm{PD}}=\frac{4}{\pi\delta^2}\frac{V_{ij}}{\left(x_j-x_i\right)^2}$. With this notation, we can write:

$$\dot{\theta}\left(\boldsymbol{x}_i,t\right)=-\sum\nolimits_j A_{ij}^{\mathrm{PD}}\left[\theta\left(\boldsymbol{x}_i,t\right)-\theta(\boldsymbol{x}_j,t)\right] \tag{17}$$

After rearranging terms, the degree of a vertex/node becomes apparent:

$$\dot{\theta}\left(\boldsymbol{x}_i,t\right)=-\theta\left(\boldsymbol{x}_i,t\right)\sum\nolimits_j A_{ij}^{\mathrm{PD}}+\sum\nolimits_j A_{ij}^{\mathrm{PD}}\theta\left(\boldsymbol{x}_j,t\right)=-\theta\left(\boldsymbol{x}_i,t\right)\deg\left(v_i\right)+\sum\nolimits_j A_{ij}^{\mathrm{PD}}\theta\left(\boldsymbol{x}_j,t\right) \tag{18}$$

where $\deg(v_i)$ is the degree of node i (see Sect. 2.1): $\deg\left(v_i\right)=\sum_j A_{ij}^{\mathrm{PD}}$. To see how this is similar to the form shown in Eq. (3), we rewrite as follows:

$$\dot{\theta}\left(\boldsymbol{x}_i,t\right)=-\sum\nolimits_j\left[\delta_{ij}\deg\left(v_i\right)-A_{ij}^{\mathrm{PD}}\right]\theta\left(\boldsymbol{x}_j,t\right)=-\sum\nolimits_j[l_{ij}^{\mathrm{PD}}]\theta(\boldsymbol{x}_j,t) \tag{19}$$

where δ_{ij} is the Kronecker delta symbol.

Based on the degree matrix and the Laplacian matrix definitions shown in Sect. 2.1, $\delta_{ij}\deg\left(v_i\right)$ is the (i,j) element in the (diagonal) degree matrix D, and $\delta_{ij}\deg\left(v_i\right)-A_{ij}^{\mathrm{PD}}$ is the element l_{ij}^{PD} in the Laplacian matrix. Notice that the value of element l_{ij}^{PD} depends on the horizon size and nodal discretization. To distinguish from the previous matrices used in Sect. 2.1, we use the notation L^{PD} to represent the Laplacian matrix obtained with peridynamics.

By writing the temperature field as a vector of nodal temperatures, $\boldsymbol{\theta}(t)=\left[\,\theta_1(t)\ \theta_2(t)\ \cdots\ \theta_{\mathrm{N}-1}(t)\ \theta_{\mathrm{N}}(t)\,\right]$, we get the following equation:

$$\frac{d\boldsymbol{\theta}(t)}{dt}=-L^{\mathrm{PD}}\boldsymbol{\theta}(t) \tag{20}$$

This equation is similar to Eq. (3) in Sect. 2.1. The exact solution of this linear first-order system of differential equations is

$$\boldsymbol{\theta}(t) = e^{-L^{\mathrm{PD}}\mathrm{t}}\boldsymbol{\theta}_0 \tag{21}$$

where $\boldsymbol{\theta}_0$ is a vector that contains the initial temperature at all nodes. In bond-based peridynamics, bond properties are symmetric: property value between nodes i and j is the same as the value for the bond between nodes j and i. Thus, the adjacency matrix A^{PD} is symmetric. Since the degree matrix D is diagonal, the Laplacian matrix, $L^{\mathrm{PD}} = D^{\mathrm{PD}} - A^{\mathrm{PD}}$, is also symmetric.

Taking the same steps as shown in Sect. 2.1, by performing eigen-decomposition on L^{PD} and using Taylor series expansion about zero, the temperature field $\boldsymbol{\theta}(t)$ is

$$\boldsymbol{\theta}(t) = \phi^{\mathrm{PD}} e^{-\Lambda^{\mathrm{PD}}t} \phi^{\mathrm{PD}^T} \boldsymbol{\theta}_0 \tag{22}$$

where ϕ^{PD} is the eigenvector matrix whose columns are eigenvectors of L^{PD} and Λ^{PD} is the eigenvalue matrix whose diagonal elements are eigenvalues of L^{PD} [44]. We have thus shown that the GL-PD is similar to the SG approach in [12, 13]. However, these two methods are not identical because of the differences in terms of how each graph is defined (based on their corresponding spatial domain discretizations), and the calibration procedures for the parameters involved: the gain factor in SG and the micro-conductivity in GL-PD.

By considering the PD meshfree discretization as an undirected weighted graph, we can say that the new GL-PD approach can be seen as a "hybrid" between the SG and the meshfree PD discretization. Note that while we selected the constant profile for the micro-conductivity function, and uniform node discretization in the derivations shown above for GL-PD, these choices are by no means restrictive: other profiles can be selected for the micro-conductivity functions [50], and non-uniform node discretizations can also be used [64]. The differences between the SG approach and GL-PD are how one computes the weights of the graph edges, and these differences come from the two main sources mentioned at the beginning of this section: the calibration procedure and the type of discretization used. In what follows, we discuss these differences in more detail.

The calibration process is one difference between the SG approach and GL-PD. As discussed in Sect. 2.1, in the SG, one calibrates, at the discrete level, the gain factor by comparing the solution obtained with Eq. (12) with a known solution, either an exact solution or one obtained with an alternative numerical method, like the FEM. The calibration is problem dependent, discretization dependent, and also depends on the truncation time considered to be close enough to the steady-state. It also depends on the location in the domain at which the error is measured in determining the value of the gain factor. In contrast, in GL-PD one calibrates, at the continuum level, the micro-conductivities by enforcing a match between the peridynamic flux to the classical heat-flux for a linear temperature field, in the bulk. In this way, the GL-PD model has linear consistency, in the bulk. This guarantees convergence to the classical solution in the limit of the horizon (δ-convergence) going to zero, for any domain or given boundary conditions [28, 50].

Another difference between the SG approach and GL-PD is in terms of how weights in the graph are defined. The GL-PD is more physically based than SG because the nodes have volumes associated with them, and the units for micro-conductivity are those of conductivity per unit volume. In SG, corresponding quantities have their physical meaning obscured into the least square minimization procedure. The gain factor in SG changes

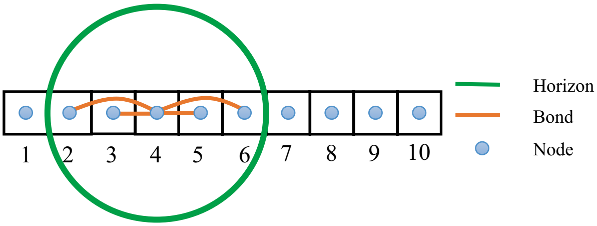


Fig. 6 Discretization of a 1D peridynamic bar

under different node discretizations, and their associated "volumes" are implicitly, not explicitly, influencing the gain factor value. To see the implications of these different discretizations and definitions of weights, consider a given randomly generated node distribution in a cube. In GL-PD, each node has an associated volume with it, and nodes in regions of slightly lower density have larger volumes. This is then reflected in the weights in the graph (elements in the adjacency matrix are defined using these volumes, see Eq. (16)). On the other hand, in SG, given the definition of the adjacency matrix elements (see Sect. 2.1), this "directionality" of low or high density is lost. If the exact solution is a constant temperature, for example, the error from the SG will be higher than that from GL-PD, by the nature of the solution method. In other words, with SG one has to use either a uniform node distribution or a random distribution which has an almost uniform spatial node density distribution, since otherwise the method does not satisfy linear consistency.

In the calibration process of GL-PD, the material points are assumed to have a full horizon. By applying the calibrated micro-conductivities to material points that do not have a full horizon (e.g., points on a boundary), the effective behavior at these points will be different. This is called the PD surface effect, which can be reduced or eliminated using PD surface correction methods in [61, 51]. However, the calibration process in SG does not consider the surface effect, as the gain factor is a single value for the entire domain. Although the calibration process of SG is different from PD, it would be interesting to investigate the behavior of SG solution near the boundaries, especially since SG has been used in additive manufacturing applications in which the region of interest is always at the boundary. This is planned for the future.

4 Limitations of the Spectral-Based Semi-analytical Approach for Diffusion Problems

In this section, we discuss some limitations the spectral-based semi-analytical approach for diffusion problem has. The semi-analytical solution in Eqs. (12) and (22) is only feasible for a homogeneous linear diffusion equation with insulated (Neumann-type) boundary conditions (see next section for an extension to Dirichlet boundary conditions). For nonlinear diffusion equations, like when material parameters (ρ, c, κ) depend on temperature, the bond micro-conductivities may vary with time. Consequently, the Laplacian matrix, its eigenvectors, and its eigenvalues may be different through time, requiring their re-computation (which costs $O(N^3)$) and thus leading to an inefficient algorithm. Thus, the spectral-based semi-analytical approach is not feasible for nonlinear diffusion equations, such as filtration problems [65], turbulent diffusion [66], corrosion damage [36], fracture [67], etc.

To show why the spectral-based semi-analytical approach in Sect. 2.1 is limited to insulated boundary conditions, here we use a simple 1D transient heat transfer problem. We shall see that the Laplacian matrix under Dirichlet boundary conditions is not symmetric.

In this sample problem, the length of the bar is 1 m, the node discretization is 0.1 m, and the horizon size is chosen to be 0.25 m. The problem domain after discretization is shown below (see Fig. 6).

We choose to construct the Laplacian matrix using the steps in Sect. 2.2, due to its simplicity in calibration. For simplicity, we set the values of ρ, c, κ to be equal to one. We use the constant micro-conductivity function with $k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{1}{\delta}$ (1D) [28]. After construction, the matrix form corresponding to Eq. (20) is

$$\frac{\partial}{\partial t}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_9 \\ \theta_{10} \end{bmatrix} = -\begin{bmatrix} 50 & -40 & -10 & 0 & 0 & \dots & 0 & 0 & 0 \\ -40 & 90 & -40 & -10 & 0 & \dots & 0 & 0 & 0 \\ -10 & -40 & 100 & -40 & -10 & \dots & 0 & 0 & 0 \\ 0 & -10 & -40 & 100 & -40 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -40 & 90 & -40 \\ 0 & 0 & 0 & 0 & 0 & \dots & -10 & -40 & 50 \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_9 \\ \theta_{10} \end{bmatrix} \tag{23}$$

In this discretization, the insulated boundary condition is satisfied implicitly.

For a problem with Dirichlet boundary conditions at both ends, the values for θ_1 and θ_{10} are known and the matrix form of Eq. (20) is

$$\frac{\partial}{\partial t}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_9 \\ \theta_{10} \end{bmatrix} = -\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ -40 & 90 & -40 & -10 & 0 & \dots & 0 & 0 & 0 \\ -10 & -40 & 100 & -40 & -10 & \dots & 0 & 0 & 0 \\ 0 & -10 & -40 & 100 & -40 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -40 & 90 & -40 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_9 \\ \theta_{10} \end{bmatrix} \tag{24}$$

As shown in Eq. (24), the matrix L is no longer symmetric. In such a case, the eigenvectors of matrix L may not be orthogonal and the spectral form in Eq. (22) is no longer applicable.

Moreover, if a non-zero Neumann boundary condition is applied to the boundary nodes, the heat flux term needs to be added to the right hand side of Eq. (23) and the solution in Eq. (4) does not hold. Therefore, the application area of this procedure is limited to homogeneous linear diffusion with insulated boundary conditions.

In addition, one has to consider that computing eigenvalues and eigenvectors carries the computational cost of $O(N^3)$. If N is very large, this cost is prohibitive. For such cases, the direct integration approach will be the only option since, for example, at each time step, the computational cost for the one point Gaussian quadrature is $O(N^2)$ and the cost of the fast convolution-based method is $O(N \log N)$.

5 Extension of the Spectral-Based Semi-analytical Solution for Graph Laplacian to Problems with Dirichlet BCs

In this section, we will show for the first time how to modify the semi-analytical approach shown in Sect. 2.1 to solve transient heat transfer problems with Dirichlet boundary conditions, for which symmetry of the Laplacian matrix is lost and the spectral decomposition is not available. The semi-analytical method was previously limited to linear diffusion problems with homogeneous Neumann boundary conditions. In our extension, we apply the Dirichlet boundary conditions using the “naïve” fictitious node method [50]. A more accurate method to impose local boundary conditions in at PD model is the “mirror-based” fictitious node method [30], but that would not work here because those conditions change with time (as the solution inside the domain changes in time), and then one can no longer use the semi-analytical approach. One could also use the direct imposition of the Dirichlet conditions only at the nodes at the surface, but in that case, the differences compared with the classical solutions are larger.

Although the semi-analytical solution in Eq. (22) is not suitable for problems with Dirichlet boundary conditions, one can still obtain a solution to the problem using Eq. (21) and computing its solution via, for example, the matrix exponential function in Matlab [68], “expm(A)”. This function is based on a scaling and squaring algorithm with a Padé approximation [69, 70]. In this function, the matrix A is scaled by a power of 2 to reduce the matrix norm to a value in the order of 1, then one computes a Padé approximation to the scaled matrix, and repeatedly square it to eliminate the effect of the scaling. The approximation $e^A \approx (r_m(2^{-s}A))^{2^s}$, where $r_m(x) = p_m(x)/q_m(x)$ is the Padé approximant to e^x, and the choice of integers m and s are discussed in detail in [69, 70]. We solved two numerical examples in 1D and 2D using this procedure.

5.1 Example 1: A 1D Bar with Dirichlet Boundary Conditions

We consider a 1D transient heat transfer problem with Dirichlet boundary conditions for a bar of length $L = 1$ m with an initial temperature of $\theta(x, 0) = 2$ °C. Both ends of the bar are maintained at 1 °C. For simplicity, here we set the values of ρ, c, κ to equal one.

Using separation of variables [43], the analytical solution [71] of the classical heat equation for this problem is

$$\begin{aligned}\theta(x,t) &= \theta(0,t) + [\theta(L,t) - \theta(0,t)]\frac{x}{L} + \sum\nolimits_{n=1}^{\infty} D_n \sin\left(\frac{n\pi x}{L}\right)e^{-\frac{n^2\pi^2\alpha t}{L^2}} \\ D_n &= \frac{2}{L}\int_0^L \theta(x,0)\sin\left(\frac{n\pi x}{L}\right)dx\end{aligned} \tag{25}$$

where $\alpha = \frac{\kappa}{\rho c}$ is the thermal diffusivity. In the numerical results shown below, we truncate the series after $n = 100$.

To solve the 1D problem using Eq. (21), the domain is uniformly discretized with 200 nodes, and the horizon size is set to 0.05 m. The Laplacian matrix is then obtained using the GL-PD method (see Sect. 3). To apply the Dirichlet boundary conditions, we use the naïve fictitious nodes method [50] by assigning the boundary conditions values to all fictitious points corresponding to a boundary point, as shown in Fig. 7.

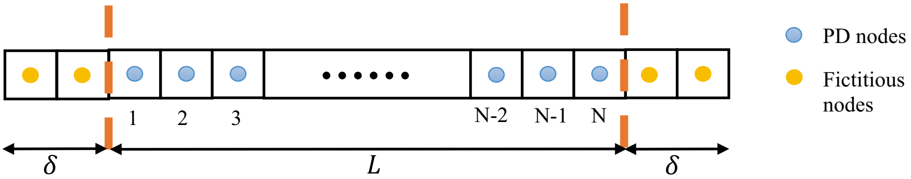


Fig. 7 The discretized bar and fictitious nodes at both ends

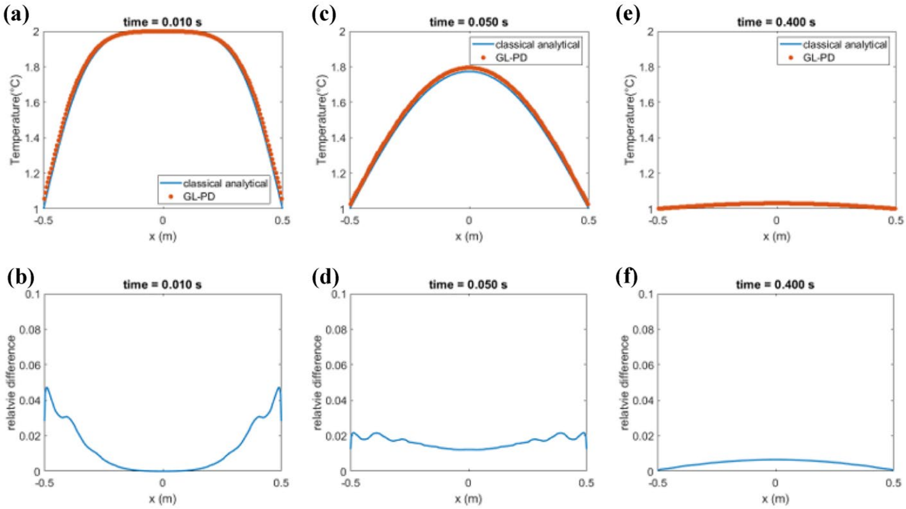


Fig. 8 The analytical solution for the classical heat transfer problem and the GL-PD solution at (**a**) 0.01 s, (**c**) 0.05 s, (**e**) 0.4 s. Relative differences between these solutions at (**b**) 0.01 s, (**d**) 0.05 s, (**f**) 0.4 s

The temperature profiles and point-wise relative differences between the analytical solution of the classical problem and the solution from GL-PD at different times are given in Fig. 8. Since we are comparing the solutions of a nonlocal model to that of the local model, the term ‘relative difference’ is used instead of ‘error’. The results show that the relative difference is less than 5%, similar to what was seen for the problem with homogeneous Neumann boundary conditions in previous studies [12, 13].

We now test the rate of convergence (δ-convergence, see Sect. 2.2) for the GL-PD and compare it with that of the numerically integrated (forward Euler) meshfree PD. The naïve fictitious node method is used in both methods to apply the boundary conditions. The relative differences between these two methods (as the horizon size decreases) and the classical analytical solution are plotted below. The temperature at the middle of the

Fig. 9 The δ convergence for the numerically integrated meshfree PD and the GL-PD

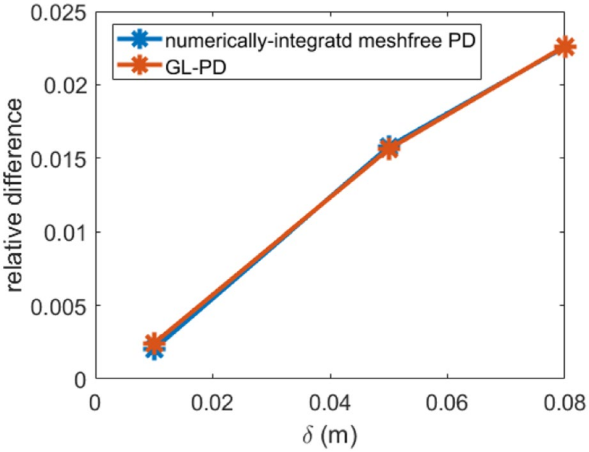
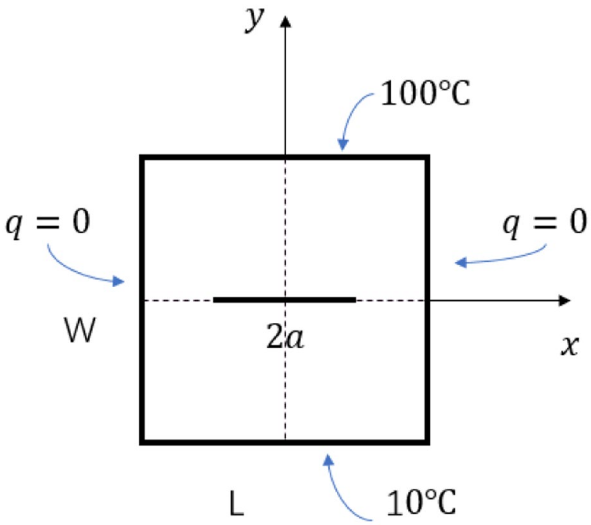


Fig. 10 Problem setup for a plate with an insulating crack



bar at 0.05 s is used to calculate the relative differences. As shown in Fig. 9, the two solution methods have very similar rates of δ-convergence.

5.2 Example 2: A 2D Plate with an Insulated Crack and Dirichlet and Neumann Boundary Conditions

To demonstrate the two-dimensional capability of the GL-PD method, we solve the problem of heat flow in a plate with an insulated crack, previously considered in [29] and solved there with the time-integrated meshfree PD. Results are compared with those for the corresponding classical model solved using ABAQUS.

A square plate, length $L = 2\,\mathrm{cm}$ and width $W = 2\,\mathrm{cm}$, has a single insulated horizontal crack of length $2a$ ($a = 0.5\,\mathrm{cm}$). The thermal diffusivity is 1.14 cm^2/s. The initial temperature of the plate is 0°C. The left and right boundaries are insulated (heat flux $q = 0$ W/m^2), while the top and bottom boundaries are, respectively, kept at 100°C and 10°C. As in Sect. 5.1, fictitious nodes are used to apply the Dirichlet boundary conditions (see Fig. 10).

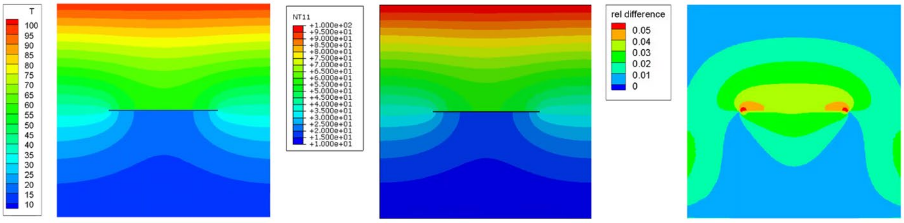


Fig. 11 The temperature distribution at time $t = 0.5$ s for GL-PD (left), for the classical model (middle), and the relative difference (right)

In the solution given by ABAQUS, a total of 6,595 linear quadrilateral elements are used. In the GL-PD solution, the domain is uniformly discretized with 10,000 nodes, the horizon size is 0.1 cm, and the m value is 5. Temperature results and the relative difference between the two solution methods at $t = 0.5$ s are shown in Fig. 11. The relative difference is computed by $\left|\theta^{\mathrm{PD}} - \theta^{\mathrm{FEM}}\right|/\left|\theta^{\mathrm{FEM}}\right|$. The mean value of relative difference at all nodes is 0.0124. Note that the classical solution, for this example, around the insulated crack tips blows up (as the mesh density increases) in terms of the temperature gradient. For a fixed horizon size in the PD formulation, the heat flux at the tips does not blow up, as the mesh is refined. This is one of the reasons for the larger differences noticed between the two solutions near the crack tips.

6 Conclusion and Future Work

In this work, we found strong connections between some recent “spectral graph” (SG) approaches used for diffusion problems (based on graph Laplacian) and the meshfree peridynamic (PD) model for diffusion. We showed that one can use the one-point Gaussian spatial integration for the peridynamic diffusion model to obtain a graph. Based on that graph, one can then define the Laplacian matrix and use the spectral-based semi-analytical approach to find a solution. We call this method the “graph Laplacian PD” (GL-PD) method. The GL-PD method can be seen as a “hybrid” between the SG and PD with meshfree discretization. While the GL-PD is similar to SG, these two solution methods differ from one another in terms of how the entries in the Laplacian matrix are defined. These differences are induced by the different calibration methods and the different discretizations used: The GL-PD is more physically based than the SG because the nodes have volumes associated with them, and the units for micro-conductivity are those of conductivity per unit volume, whereas in SG, corresponding quantities have their physical meaning obscured into the least square minimization procedure.

The SG method is limited to solving linear and homogeneous diffusion problems with insulated boundary conditions. Here, we extended the semi-analytical approach (and use it in the GL-PD context) to problems with Dirichlet boundary conditions. We apply the Dirichlet boundary conditions using the “naïve” fictitious nodes method and replace the spectral decomposition method with a scaling and squaring algorithm with a Padé approximation to calculate the matrix exponential for non-symmetric matrices. Results showed that the semi-analytical method has the same δ-convergence properties as the meshfree discretization with direct time-integration (forward Euler) of the PD model.

In the future, we will extend the graph Laplacian method to include non-zero Neumann and Robin boundary conditions. This study showed that peridynamic models may potentially be used to solve problems in graph theory, like graph classification, community detection, etc.

Funding This work was supported by the National Science Foundation under CDS&E CMMI award No. 1953346.

References

1. Ghosh S, Das N, Gonçalves T, Quaresma P, Kundu M (2018) The journey of graph kernels through two decades. Comput Sci Rev 27:88–111. https://doi.org/10.1016/j.cosrev.2017.11.002
2. Gasteiger J, Thomas Engel E (2003) *Chemoinformatics: A Textbook*. https://doi.org/10.1002/3527601643.fmatter

3. Xenarios I, Salwínski L, Duan XJ, Higney P, Kim S-M, Eisenberg D (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res 30(1):303–305. https://doi.org/10.1093/nar/30.1.303
4. Wasserman S, Faust K (1994) Social network analysis: Methods and applications. Cambridge University Press. https://doi.org/10.1017/CBO9780511815478
5. Harchaoui Z, Bach F (2007) Image classification with segmentation graph kernels. IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 2007:1–8. https://doi.org/10.1109/CVPR.2007.383049
6. Weis M, Naumann F (2006) Detecting Duplicates in Complex XML Data. *22nd International Conference on Data Engineering (ICDE'06)*, 109. https://doi.org/10.1109/ICDE.2006.49
7. Das N, Ghosh S, Gonçalves T, Quaresma P (2014) Using Graphs and Semantic Information to Improve Text Classifiers, pp: 324–336, in: Przepiórkowski A., Ogrodniczuk M. (eds) Advances in Natural Language Processing. NLP 2014. Lecture Notes in Computer Science, vol 8686. Springer, Cham. https://doi.org/10.1007/978-3-319-10888-9_33
8. Debnath AK, Lopez de Compadre RL, Debnath G, Shusterman AJ, Hansch C (1991) Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. J Med Chem 34(2):786–797. https://doi.org/10.1021/jm00106a046
9. Kondor RI, Lafferty JD (2002) Diffusion kernels on graphs and other discrete input spaces. Proceedings of the Nineteenth International Conference on Machine Learning: 315–322.
10. Lafferty J, Lebanon G (2005) Diffusion kernels on statistical manifolds. J Mach Learn Res 6:129–163. https://doi.org/10.1184/r1/6604859.v1
11. Kloster K, Gleich DF (2014) Heat kernel based community detection, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 1386–1395. https://doi.org/10.1145/2623330.2623706
12. Reza Yavari M, Cole KD, Rao P (2019) Thermal modeling in metal additive manufacturing using graph theory. J Manuf Sci Eng 141(7):071007. https://doi.org/10.1115/1.4043648
13. Cole KD, Yavari MR, Rao PK (2020) Computational heat transfer with spectral graph theory: quantitative verification. Int J Therm Sci 153:106383. https://doi.org/10.1016/j.ijthermalsci.2020.106383
14. Silling SA (2000) Reformulation of elasticity theory for discontinuities and long-range forces. J Mech Phys Solids 48(1):175–209. https://doi.org/10.1016/S0022-5096(99)00029-0
15. Bobaru F, Zhang G (2015) Why do cracks branch? A peridynamic investigation of dynamic brittle fracture. Int J Fract 196(1–2):59–98. https://doi.org/10.1007/s10704-015-0056-8
16. Mehrmashhadi J, Wang L, Bobaru F (2019) Uncovering the dynamic fracture behavior of PMMA with peridynamics: The importance of softening at the crack tip. Eng Fract Mech 219:106617. https://doi.org/10.1016/j.engfracmech.2019.106617
17. Huang D, Lu G, Qiao P (2015) An improved peridynamic approach for quasi-static elastic deformation and brittle fracture analysis. Int J Mech Sci 94:111–122. https://doi.org/10.1016/j.ijmecsci.2015.02.018
18. Lipton R (2014) Dynamic brittle fracture as a small horizon limit of peridynamics. J Elast 117(1):21–50. https://doi.org/10.1007/s10659-013-9463-0
19. Karpenko O, Oterkus S, Oterkus E (2020) Influence of different types of small-size defects on propagation of macro-cracks in brittle materials. Journal of Peridynamics and Nonlocal Modeling 2(3):289–316. https://doi.org/10.1007/s42102-020-00032-z
20. Liu PF, Li XK (2019) Dynamic void growth and localization behaviors of glassy polymer using nonlocal explicit finite element analysis. Journal of Peridynamics and Nonlocal Modeling 1(1):3–13. https://doi.org/10.1007/s42102-018-0003-y
21. Behzadinasab M, Foster JT (2019) The third Sandia Fracture Challenge: peridynamic blind prediction of ductile fracture characterization in additively manufactured metal. Int J Fract 218(1):97–109. https://doi.org/10.1007/s10704-019-00363-z
22. Yolum U, Taştan A, Güler MA (2016) A peridynamic model for ductile fracture of moderately thick plates. Procedia Structural Integrity 2:3713–3720. https://doi.org/10.1016/j.prostr.2016.06.461
23. Amani J, Oterkus E, Areias P, Zi G, Nguyen-Thoi T, Rabczuk T (2016) A non-ordinary state-based peridynamics formulation for thermoplastic fracture. Int J Impact Eng 87:83–94. https://doi.org/10.1016/j.ijimpeng.2015.06.019
24. Hu W, Ha YD, Bobaru F (2011) Modeling dynamic fracture and damage in a fiber-reinforced composite lamina with peridynamics. Int J Multiscale Comput Eng 9(6):707–726. https://doi.org/10.1615/IntJMultCompEng.2011002651
25. Hu W, Ha YD, Bobaru F (2012) Peridynamic model for dynamic fracture in unidirectional fiber-reinforced composites. Comput Methods Appl Mech Eng 217:247–261. https://doi.org/10.1016/j.cma.2012.01.016

26. Gao Y, Oterkus S (2019) Fully coupled thermomechanical analysis of laminated composites by using ordinary state based peridynamic theory. Compos Struct 207:397–424. https://doi.org/10.1016/j.compstruct.2018.09.034
27. Hu YL, Madenci E (2017) Peridynamics for fatigue life and residual strength prediction of composite laminates. Compos Struct 160:169–184. https://doi.org/10.1016/j.compstruct.2016.10.010
28. Bobaru F, Duangpanya M (2010) The peridynamic formulation for transient heat conduction. Int J Heat Mass Transf 53(19–20):4047–4059. https://doi.org/10.1016/j.ijheatmasstransfer.2010.05.024
29. Bobaru F, Duangpanya M (2012) A peridynamic formulation for transient heat conduction in bodies with evolving discontinuities. J Comput Phys 231(7):2764–2785. https://doi.org/10.1016/j.jcp.2011.12.017
30. Oterkus S, Madenci E, Agwai A (2014) Peridynamic thermal diffusion. J Comput Phys 265: 71-96. https://doi.org/10.1016/j.jcp.2014.01.027
31. Wang L, Xu J, Wang J (2018) A peridynamic framework and simulation of non-Fourier and nonlocal heat conduction. Int J Heat Mass Transf 118:1284–1292. https://doi.org/10.1016/j.ijheatmasstransfer.2017.11.074
32. Diyaroglu C, Oterkus S, Oterkus E, Madenci E (2017) Peridynamic modeling of diffusion by using finite-element analysis. IEEE Transactions on Components, Packaging and Manufacturing Technology 7(11):1823–1831. https://doi.org/10.1109/TCPMT.2017.2737522
33. Jafarzadeh S, Chen Z, Li S, Bobaru F (2019) A peridynamic mechano-chemical damage model for stress-assisted corrosion. Electrochim Acta 323:134795. https://doi.org/10.1016/j.electacta.2019.134795
34. De Meo D, Oterkus E (2017) Finite element implementation of a peridynamic pitting corrosion damage model. Ocean Eng 135:76–83. https://doi.org/10.1016/j.oceaneng.2017.03.002
35. Chen Z, Bobaru F (2015) Peridynamic modeling of pitting corrosion damage. J Mech Phys Solids 78:352–381. https://doi.org/10.1016/j.jmps.2015.02.015
36. Jafarzadeh S, Chen Z, Zhao J, Bobaru F (2019) Pitting, lacy covers, and pit merger in stainless steel: 3D peridynamic models. Corros Sci 150:17–31. https://doi.org/10.1016/j.corsci.2019.01.006
37. Bobaru F, Yang M, Alves LF, Silling SA, Askari E, Xu J (2009) Convergence, adaptive refinement, and scaling in 1D peridynamics. Int J Numer Meth Eng 77(6):852–877. https://doi.org/10.1002/nme.2439
38. Silling SA, Askari E (2005) A meshfree method based on the peridynamic model of solid mechanics. Comput Struct 83(17–18):1526–1535. https://doi.org/10.1016/j.compstruc.2004.11.026
39. Deo N (2017) *Graph theory with applications to engineering and computer science*. Courier Dover Publications.
40. Bıyıkoglu T, Leydold J, Stadler PF (2007) Laplacian eigenvectors of graphs. Lecture notes in mathematics, 1915.
41. Merris R (1994) Laplacian matrices of graphs: a survey. Linear Algebra Appl 197:143–176.
42. Carslaw HS, Jaeger JC (1992) *Conduction of heat in solids*. Clarendon press.
43. Boyce WE, DiPrima RC, Meade DB (2017) Elementary differential equations. John Wiley & Sons
44. Greub WH (2012) *Linear algebra*. Springer Science & Business Media.
45. Hammond DK, Gur Y, Johnson CR (2013) Graph diffusion distance: a difference measure for weighted graphs based on the graph Laplacian exponential kernel. IEEE Global Conference on Signal and Information Processing 2013:419–422. https://doi.org/10.1109/GlobalSIP.2013.6736904
46. Fouss F, Francoisse K, Yen L, Pirotte A, Saerens M (2012) An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. Neural Netw 31:53–72. https://doi.org/10.1016/j.neunet.2012.03.001
47. Chen Z, Niazi S, Bobaru F (2019) A peridynamic model for brittle damage and fracture in porous materials. Int J Rock Mech Min Sci 122:104059. https://doi.org/10.1016/j.ijrmms.2019.104059
48. Jafarzadeh S, Chen Z, Bobaru F (2018) Peridynamic modeling of intergranular corrosion damage. J Electrochem Soc 165(7):C362. https://doi.org/10.1149/2.0821807jes
49. Zhao J, Chen Z, Mehrmashhadi J, Bobaru F (2018) Construction of a peridynamic model for transient advection-diffusion problems. Int J Heat Mass Transf 126:1253–1266. https://doi.org/10.1016/j.ijheatmasstransfer.2018.06.075
50. Chen Z, Bobaru F (2015) Selecting the kernel in a peridynamic formulation: a study for transient heat diffusion. Comput Phys Commun 197:51–60. https://doi.org/10.1016/j.cpc.2015.08.006
51. Prakash N (2019) Calibrating bond-based peridynamic parameters using a novel least squares approach. Journal of Peridynamics and Nonlocal Modeling 1(1):45–55. https://doi.org/10.1007/s42102-018-0002-z

52. Chen Z, Ju JW, Su G, Huang X, Li S, Zhai L (2019) Influence of micro-modulus functions on peridynamics simulation of crack propagation and branching in brittle materials. Eng Fract Mech 216:106498. https://doi.org/10.1016/j.engfracmech.2019.106498
53. Chen X, Gunzburger M (2011) Continuous and discontinuous finite element methods for a peridynamics model of mechanics. Comput Methods Appl Mech Eng 200(9–12):1237–1250. https://doi.org/10.1016/j.cma.2010.10.014
54. Du Q, Tian L, Zhao X (2013) A convergent adaptive finite element algorithm for nonlocal diffusion and peridynamic models. SIAM J Numer Anal 51(2):1211–1234. https://doi.org/10.1137/120871638
55. Jafarzadeh S, Larios A, Bobaru F (2020) Efficient solutions for nonlocal diffusion problems via boundary-adapted spectral methods. J Peridyn Nonlocal Model 2, 85–110. https://doi.org/10.1007/s42102-019-00026-6
56. JafarzadehWang L, Larios A, Bobaru FS (2021) A fast convolution-based method for peridynamic transient diffusion in arbitrary domains. Comput Methods Appl Mech Engrg 375:113633. https://doi.org/10.1016/j.cma.2020.113633
57. Murakami S (2012) *Continuum damage mechanics: a continuum mechanics approach to the analysis of damage and fracture*. Springer Science & Business Media.
58. Silling SA, Epton M, Weckner O, Xu J, Askari E (2007) Peridynamic states and constitutive modeling. J Elast 88(2):151–184. https://doi.org/10.1007/s10659-007-9125-1
59. Wu P, Zhao J, Chen Z, Bobaru F (2020) Validation of a stochastically homogenized peridynamic model for quasi-static fracture in concrete. Eng Fract Mech 237:107293. https://doi.org/10.1016/j.engfracmech.2020.107293
60. Bobaru F, Ha YD, Hu W (2012) Damage progression from impact in layered glass modeled with peridynamics. Central Eur J Eng 2(4):551–561. https://doi.org/10.2478/s13531-012-0020-6
61. Le QV, Bobaru F (2018) Surface corrections for peridynamic models in elasticity and fracture. Comput Mech. 61: 499–518. https://doi.org/10.1007/s00466-017-1469-1
62. Bobaru F, Ha YD (2011) Adaptive refinement and multiscale modeling in 2D peridynamics. Int J Multiscale Comput Eng 9(6):635–659. https://doi.org/10.1615/IntJMultCompEng.2011002793
63. Gu X, Zhang Q, Xia X (2017) Voronoi-based peridynamics and cracking analysis with adaptive refinement. Int J Numer Meth Eng 112(13):2087–2109. https://doi.org/10.1002/nme.5596
64. Henke SF, Shanbhag S (2014) Mesh sensitivity in peridynamic simulations. Comput Phys Commun 185(1):181–193. https://doi.org/10.1016/j.cpc.2013.09.010
65. Huilai L, Zhuoqun W, Jingxue Y, Junning Z (2001) Nonlinear diffusion equations. World Scientific
66. Csanady GT (2012) Turbulent diffusion in the environment. Springer Science & Business Media.
67. Chen Z, Jafarzadeh S, Zhao J, Bobaru F (2021) A coupled mechano-chemical peridynamic model for pit-to-crack transition in stress-corrosion cracking. J Mech Phys Solids 146:104203. https://doi.org/10.1016/j.jmps.2020.104203
68. MATLAB (2020) Natick. The MathWorks Inc, Massachusetts
69. Higham NJ (2005) The scaling and squaring method for the matrix exponential revisited. SIAM J Matrix Anal Appl 26(4):1179–1193. https://doi.org/10.1137/04061101X
70. Al-Mohy AH, Higham NJ (2010) A new scaling and squaring algorithm for the matrix exponential. SIAM J Matrix Anal Appl 31(3):970–989. https://doi.org/10.1137/09074721X
71. Greenberg MD (1988) Advanced Engineering Mathematics. Prentice-Hall