Connections between peridynamics and graph Laplacian for diffusion problems

Longzhen Wang[a], Florin Bobaru[a*]

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

*Corresponding author. Email address: fbobaru2@unl.edu

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 discretizations for corresponding peridynamic models have a similar graph structure and lead to the same general equation as the SG approach. In this sense, the SG approach can be seen as a particular case of a peridynamic formulation. For the transient heat diffusion, we explain some differences between the SG approach and peridynamics, 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 peridynamic models. We also introduce an extension of the semi-analytical approach to diffusion problems with Dirichlet boundary conditions.

Keywords: peridynamics, graph theory, graph Laplacian, spectral graph, heat transfer, diffusion

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], natural languages [7]. In chemistry and biology, identifying the 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 unfordable 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 can be seen as a particular case of 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][16], fracture and damage in composites [17][18], heat transfer [19][20], corrosion damage [21][22], 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) [23][24] or heat-flux under constant thermal gradient conditions [19]. 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 [24]. 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 Section 2.1. The peridynamic diffusion model for diffusion is reviewed in Section 2.2. In Section 3, using a 2D transient heat transfer problem as an example, we show that the SG approach is formally identical to a PD model solved by spectral-based semi-analytical approach. Differences related to the calibration and discretization procedures are highlighted. In Section 4, a numerical example for 1D transient heat transfer is used to show limitations of the spectral-based semi-analytical approach. In Section 5, we introduce an extension of the semi-analytical approach to diffusion problems with Dirichlet boundary conditions. In Section 6, we present conclusions and future work.

2. The graph Laplacian exponential diffusion kernel and the peridynamic model for diffusion

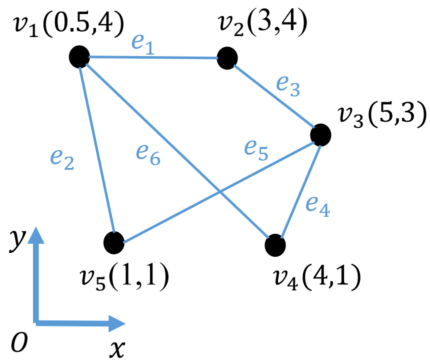
2.1. Review of graphs and the graph Laplacian exponential diffusion kernel

In mathematics, a graph consists of a set of vertices $V = \{v_1, v_2, \ldots\}$ and another set of edges $E = \{e_1, e_2, \ldots\}$ such that each edge is identified with an unordered pair of vertices [25]. Graphs are commonly represented as diagrams in which vertices are nodes and edges are represented by lines connecting nodes [25], as shown in Figure 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 [25]. There are different matrices used in graph theory [25]; here we focus only on the adjacency and Laplacian matrices.

The adjacency matrix of an N-vertex weighted undirected simple graph is an N by N real-symmetric matrix $A = [a_{ij}]$, with a_{ij} = weight of edge between node i and node 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 Figure 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 weight of this edge is 0.16.

The degree matrix of an N-vertex weighted undirected simple graph is an N by N diagonal matrix $D = [d_{ij}]$, with d_{ii} equals the degree of i^{th} vertex which is the sum of the weight 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 Figure 1.



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

$$D = \begin{bmatrix} 0.3152 & 0 & 0 & 0 & 0 \\ 0 & 0.36 & 0 & 0 & 0 \\ 0 & 0 & 0.45 & 0 & 0 \\ 0 & 0 & 0 & 0.2471 & 0 \\ 0 & 0 & 0 & 0 & 0.1581 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.3152 & -0.16 & 0 & -0.0471 & -0.1081 \\ -0.16 & 0.36 & -0.2 & 0 & 0 \\ 0 & -0.2 & 0.45 & -0.2 & -0.05 \\ -0.0471 & 0 & -0.2 & 0.2471 & 0 \\ -0.1081 & 0 & -0.05 & 0 & 0.1581 \end{bmatrix}$$

Figure 1 Undirected Graph (v: vertices, e: edges) and its corresponding adjacency matrix, degree matrix, and Laplacian matrix

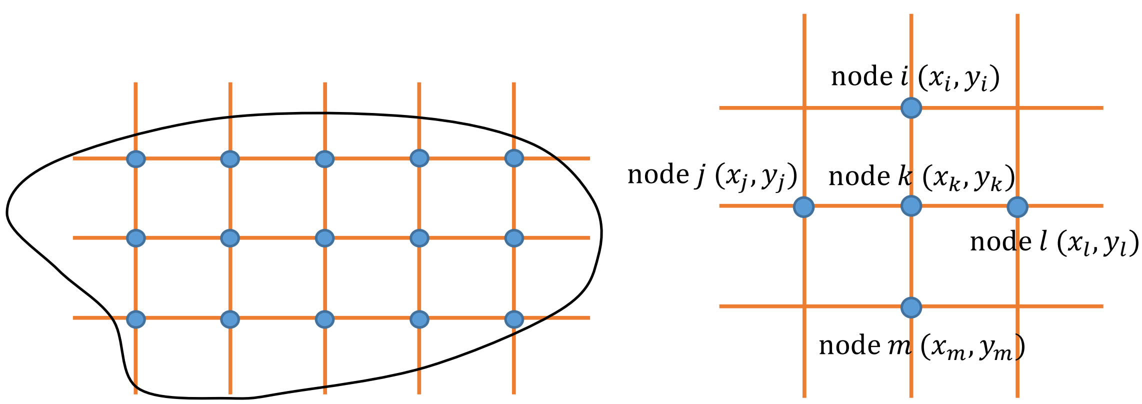


Figure 2 A domain discretized by a uniform grid (left), and the 5-point stencil for finite difference approximations (right).

The negative Laplacian matrix can be seen as a proper discretization of the classical Laplacian operator [26][27]. For example, on a square grid in 2D. with grid spacing $\Delta x = \Delta y = h$, the finite difference approximation of the continuous Laplace operator at node k, of coordinates (x_k, y_k), is $\frac{\partial^2\theta}{\partial x^2}(x_k, y_k) + \frac{\partial^2\theta}{\partial y^2}(x_k, y_k) \approx \frac{\theta_l - 2\theta_k + \theta_j}{(\Delta x)^2} + \frac{\theta_i - 2\theta_k + \theta_m}{(\Delta y)^2} = \frac{1}{h^2}(\theta_l + \theta_j + \theta_i + \theta_m - 4\theta_k)$. If the square grid is seen as a graph, node k is connected to four neighboring nodes (node i, node j, node l, and node m). In the graph, the edge weight is the inverse of edge length square, same as Figure 1. The continuous Laplacian at node k can then be approximated by the negative Laplacian matrix multiplied by vector $\boldsymbol{\theta}$, which leads to: $\frac{1}{h^2}(\theta_l + \theta_j + \theta_i + \theta_m - 4\theta_k)$, (see Figure 3). Thus, $-L$ is the same as the finite difference approximation of the continuous Laplacian operator [9]. In the limit $h \rightarrow 0$, this approximation becomes exact [9].

$$\underset{N \times 1\ vector}{\begin{bmatrix} \vdots \\ \frac{\partial^2 \theta}{\partial x^2}(x_k, y_k) + \frac{\partial^2 \theta}{\partial y^2}(x_k, y_k) \\ \vdots \end{bmatrix}} = \underset{N \times N\ matrix}{\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 1/h^2 & \cdots & 1/h^2 & \cdots & -4/h^2 & \cdots & 1/h^2 & \cdots & 1/h^2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}} \underset{N \times 1\ vector}{\begin{bmatrix} \vdots \\ \theta_i \\ \vdots \\ \theta_j \\ \vdots \\ \theta_k \\ \vdots \\ \theta_l \\ \vdots \\ \theta_m \\ \vdots \end{bmatrix}}$$

Figure 3 The approximation of the continuous Laplacian at node k using the Laplacian matrix

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 which are located within a certain distance *b*, called “neighborhood distance”. 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] use 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 Figure 4.

Heat flow in 2D can be described, under some limiting conditions (see Section 4) by the following linear partial differential equation (PDE) [28]:

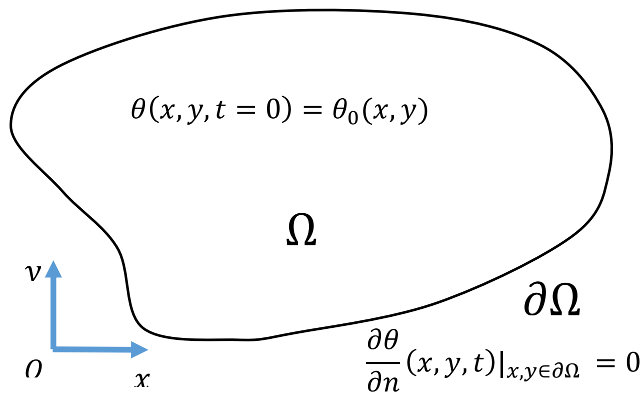


Figure 4 Domain and boundary conditions for a linear 2D transient heat transfer problem with insulated boundaries.

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

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 $(J/kg \cdot K)$, κ is the thermal conductivity $(W/m \cdot K)$, 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):

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

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 $A_{ij} = 1/c_{ij}^2, c_{ij} \le b,\ i \ne j$, 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 bell-shaped function, $A_{ij} = e^{-c_{ij}^2/\sigma^2}, c_{ij} \le b,\ i \ne j$, 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 $\boldsymbol{\theta}(t) = [\theta_1(t) \quad \theta_2(t) \quad \cdots \quad \theta_{N-1}(t) \quad \theta_N(t)]$, 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 [29]:

$$\boldsymbol{\theta}(t) = e^{-Lt}\boldsymbol{\theta}_{\mathbf{0}} \tag{4}$$

where $\boldsymbol{\theta}_{\mathbf{0}}$ is a 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 following equation:

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

For each eigenvalue λ, the corresponding eigenvector $\boldsymbol{v}$ can be found solving the system:

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

With the eigenvalues λ and eigenvectors found, by performing eigen-decomposition, 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 eigenvalue matrix whose diagonal elements are eigenvalues of L [30]. 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 ([27]), and, therefore, its eigenvectors ϕ are orthogonal [30]. 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:

$$\boldsymbol{\theta}(t) = e^{-\phi \Lambda \phi^{T} t}\boldsymbol{\theta}_{\mathbf{0}} \tag{9}$$

where the term $e^{-\phi \Lambda \phi^{T} t}$ is called the graph Laplacian exponential diffusion kernel [31][32].

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

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

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

$$\boldsymbol{\theta}(t) = \phi e^{-\Lambda t}\phi^T \boldsymbol{\theta}_{\mathbf{0}} \tag{11}$$

To make a connection between edges' weight 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]):

$$\boldsymbol{\theta}(t) = \phi e^{-g\Lambda t}\phi^T \boldsymbol{\theta}_{\mathbf{0}} \tag{12}$$

To obtain the "gain factor", a problem-dependent calibration (note the difference between the gain factors obtained in references [12] and [13]) is 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 geometry) is monitored to steady-state and the factor g that minimizes the least-squares 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), 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 Section 4.

2.2. The peridynamic model for diffusion

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

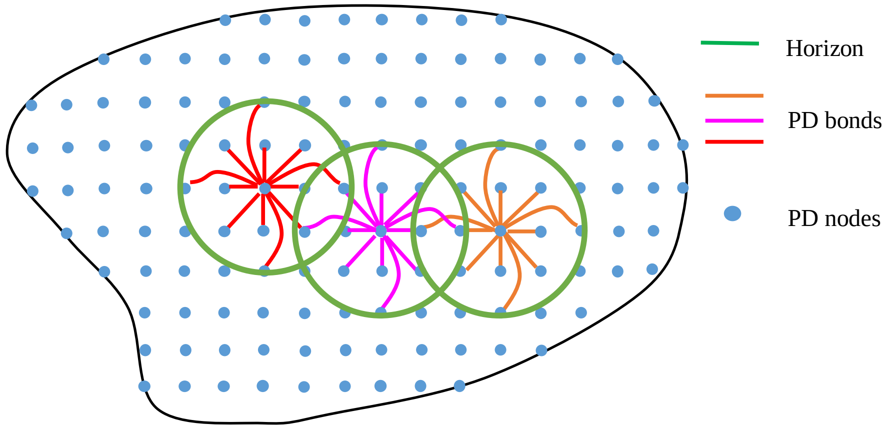


Figure 5 Schematic diagram of a peridynamic discretization (based on the one-point Gaussian quadrature in space) showing a few nodes and their bonds that create a graph structure for this PD discretization.

The bond-based peridynamic model for diffusion problem, studied in [19], 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][20]. This is the reason why peridynamic formulations are widely used to solve problems with discontinuities such as cracks [16], damage [33], dissolution [34], etc.

In peridynamics, each point is connected and interacts with all other points within a certain region, called the horizon region, as shown in Figure 5. The bond-based PD equation for diffusion is [19]:

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

where $H_{\boldsymbol{x}}$ is the horizon of $\hat{\boldsymbol{x}}$, $k(\boldsymbol{x},\hat{\boldsymbol{x}})$ is the $(\boldsymbol{x},\hat{\boldsymbol{x}})$ bond micro-conductivity $(W/m^3 \cdot K)$. Instead of using $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2$ in kernel denominator, one can also use, for example $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^1$ or $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^0$ (see [35][36]).

The peridynamic heat flux at point $\boldsymbol{x}$ across surface S is defined by [19]:

$$Q(\boldsymbol{x},S,t) = -\int_{H_{\boldsymbol{x}}/S} k(\boldsymbol{x},\hat{\boldsymbol{x}}) \frac{\theta(\hat{\boldsymbol{x}},t) - \theta(\boldsymbol{x},t)}{(\hat{\boldsymbol{x}} - \boldsymbol{x}) \cdot \boldsymbol{n}} d\hat{\boldsymbol{x}} \tag{14}$$

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

The micro-conductivity function $k(\boldsymbol{x},\hat{\boldsymbol{x}})$ describes the properties of bonds between material points inside the horizon. 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 [20]. With this calibration, and for the particular profile of a micro-conductivity function that is constant over the horizon, in 2D one obtains $k(\boldsymbol{x},\hat{\boldsymbol{x}}) = {}^{4\kappa}/_{\pi\delta^2}$ (see [19][20]), 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. [37][36]. Different micro-conductivity functions may lead to different convergence rates to the classical solution (see [37]) as the horizon δ decreases to zero while the ratio between δ and the discretization spacing is maintained constant (the so-called δ-convegence, see [23]).

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" method) [19] [23], the finite element method [38][39], the fast convolution-based method [40][41], etc. In the meshfree method, there are no elements or other geometrical connections between nodes [24]. 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 [42]). In some sense, PD damage is a mapping, not necessarily continuous, from the vector space of bonds at a point 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 [43]). 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 [38]. 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 ([44], [45]).

3. The spectral-based semi-analytical approach for the bond-based Peridynamic diffusion model

In this section, we show that by applying the spectral-based semi-analytical approach described in Section 2.1 to solve the peridynamic diffusion model, one obtains a *formally identical* equation to that found via the SG approach shown in [12] and [13]. While formally identical, some differences between the SG approach and the PD still exist:

- the calibration procedures are different: in 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 any problem with any domain geometry and boundary conditions; in SG, a calibration has to be performed for each new problem; and

- the specific definition of the terms in the adjacency matrix is different: the SG and 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 PD, nodes come with their associated volume (area in 2D, length in 1D).

With the meshfree discretization in 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 bonds are the edges in the graph, the nodes are its vertices, and a combination of bond micro-conductivities and geometrical quantities (see below) are the weights of the graph edges.

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 [46]. Note that non-uniform discretization can also be used in PD models ([47], [48], [49]). With the meshfree discretization, Eq. (13) becomes ([19]):

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

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 with the choice made in Section 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 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = {}^4\!/_{\pi\delta^2}$ (2D). The discretized form becomes (notice the switch in the numerator, useful later for matching the definition of the Laplacian matrix):

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

In the 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 $A_{ij}^{\mathrm{PD}} = \frac{k(x_i, x_j) V_{ij}}{(x_j - x_i)^2}$.

In the case of constant micro-conductivity, the weight becomes $A_{ij}^{\mathrm{PD}} = \frac{4}{\pi\delta^2} \frac{V_{ij}}{(x_j - x_i)^2}$. With this notation, we can write:

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

After rearranging terms, the degree of a vertex/node (see the degree of a vertex in Section 2.1) becomes apparent:

$$\begin{aligned} \dot{\theta}(\boldsymbol{x}_i, t) &= -\left[\theta(\boldsymbol{x}_i, t) \sum_j A_{ij}^{\mathrm{PD}} - \sum_j A_{ij}^{\mathrm{PD}} \theta(\boldsymbol{x}_j, t)\right] = \\ &= -\left[\theta(\boldsymbol{x}_i, t) \deg(v_i) - \sum_j A_{ij}^{\mathrm{PD}} \theta(\boldsymbol{x}_j, t)\right] \end{aligned} \tag{18}$$

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

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

where δ_{ij}, is the Kronecker delta symbol.

Based on the degree matrix and the Laplacian matrix definitions shown in section 2.1, $\delta_{ij} \deg(v_i)$ is the (i,j) element in the (diagonal) degree matrix D, and $\delta_{ij} \deg(v_i) - A_{ij}^{\mathrm{PD}}$ 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 Section 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, $\boldsymbol{\theta}(t) = [\theta_1(t) \quad \theta_2(t) \quad \cdots \quad \theta_{N-1}(t) \quad \theta_N(t)]$, 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 Section 2.1. The exact solution of this linear first-order system of differential equations is:

$$\boldsymbol{\theta}(t) = e^{-L^{\mathrm{PD}} t} \boldsymbol{\theta}_{\mathbf{0}} \tag{21}$$

where $\boldsymbol{\theta}_{\mathbf{0}}$ 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 symmetric.

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

$$\boldsymbol{\theta}(t) = \phi^{\mathrm{PD}} e^{-\Lambda^{\mathrm{PD}} t} {\phi^{\mathrm{PD}}}^{T} \boldsymbol{\theta}_{\mathbf{0}} \tag{22}$$

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} [30]. We have thus shown that formally, the PD with the spectral-based solution procedure is identical to the SG approach in [12], [13]. There are some differences in the following: how one performs the calibration (see the gain factor in SG versus micro-conductivities in PD) and what type of discretization is used (SG considers nodes as geometrical points, while the meshfree discretization in PD uses nodes which are representative of material volumes).

By considering the PD meshfree discretization as an undirected weighted graph and reforming the discretized peridynamic diffusion equation, we can say that, at least formally, the SG approach can be seen as a particular case of bond-based peridynamic formulation for diffusion. Note that while we selected the constant profile for the micro-conductivity function, and uniform node discretization in the derivations above, these choices are by no means restrictive: other profiles can be selected for the micro-conductivity functions [36], and non-uniform node discretizations can also be used [49]. The differences between the SG approach and 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 PD. As discussed in Section 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 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 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 problem with constant thermal gradient. 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[19][36].

Another difference between the SG approach and PD is in terms of how weights in the graph are defined. These come from the different discretizations used: SG considers nodes as geometrical points only, while in PD nodes are representative of material points, or finite regions (volumes, areas, etc.) in the domain. To see the implications of these different discretizations and definitions of weights, consider a given randomly generated node distribution in a cube. In 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 Section 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 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. Otherwise, the method does not satisfy linear consistency.

In the calibration process of 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 [46]. Although the calibration process of SG is different than 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 Eq. (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-base semi-analytical approach is not feasible for nonlinear diffusion equations, such as filtration problems [50], turbulent diffusion [51], corrosion damage [22], fracture [52], etc.

To show why the spectral-based semi-analytical approach in Section 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 Neumann and 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.

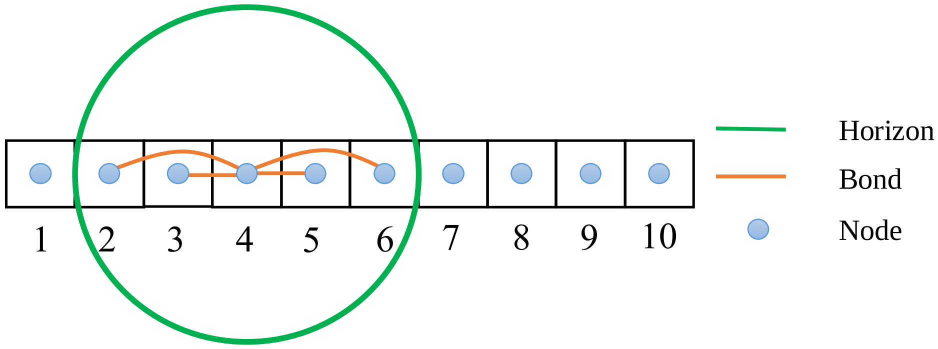


Figure 6 Discretization of a 1D peridynamic bar

We choose to construct the Laplacian matrix using the steps in Section 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 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = {}^{1}/_{\delta}$ (1D) [19]. 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, we give the values for θ_1 and θ_{10} 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 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 $O(NlogN)$.

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 Section 2.1 to solve transient heat transfer problems with Dirichlet boundary conditions. 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 [53], "expm". This function is based on a scaling and squaring algorithm with a Padé approximation [54][55].

As a numerical example for this procedure, which we will refer to as the "Padé-approximation semi-analytical approach" (PASA), we consider the 1D transient heat transfer problem with Dirichlet boundary conditions. A bar of length $L = 1$ m has 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 [29], the analytical solution [56] of the classical heat equation for this problem is:

$$\theta(x,t) = \theta(0,t) + \big(\theta(L,t) - \theta(0,t)\big)\frac{x}{L} + \sum_{n=1}^{\infty} D_n \sin(\frac{n\pi x}{L})\, e^{-\frac{n^2\pi^2\alpha t}{L^2}}$$
$$D_n = \frac{2}{L}\int_0^L \theta(x,0)\sin(\frac{n\pi x}{L})\,dx \qquad (25)$$

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 methods showed in detail in Section 3, due to its simplicity in calibration. To apply the Dirichlet boundary conditions, we use the na ïve fictitious nodes method [36] by assigning the boundary condition value to all fictitious points corresponding to a boundary point, as shown:

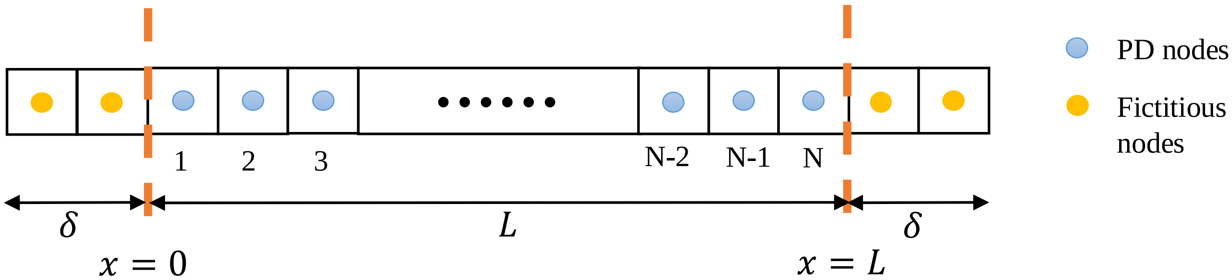


Figure 7 The discretized bar and fictitious nodes at both ends.

The temperature profiles and point-wise relative differences between the analytical solution of the classical problem and the solution from Eq. (21) at different times are given in Figure 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 zero Neumann boundary conditions in previous studies [12][13].

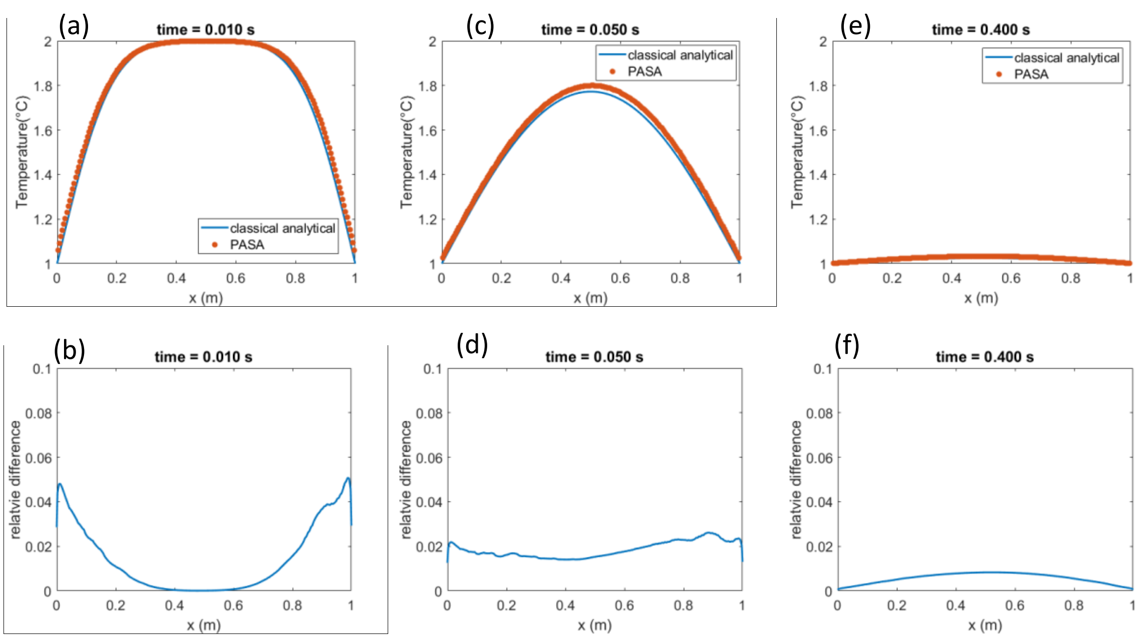


Figure 8 The analytical solution for the classical heat transfer problem and the semi-analytical PD solution at (a) 0.01s (c) 0.05s (e) 0.4s. Relative differences between these solutions at (b) 0.01s (d) 0.05s (f) 0.4s.

We now test the rate of convergence (δ-convegence, see Section 2.2) for the semi-analytical method and compare it with that of the meshfree discretization of the PD model. The naïve fictitious node method is used in both methods to apply the boundary conditions. The relative differences between these two methods and the classical analytical solution are plotted below. The temperature at the middle of the bar at 0.05s is used to calculate the relative differences.

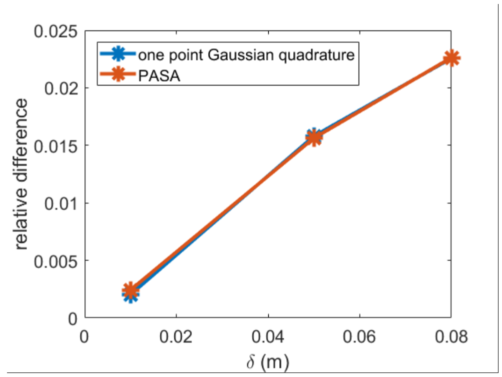


Figure 9 The δ convergence for the meshfree PD and the PASA approach for PD.

As shown in Figure 9, the two solution methods have very similar rates of δ-convegence.

6. Conclusion & 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 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 that is formally identical to that obtained by the SG approach. From this point of view, the SG approach can then be considered as a particular case of a peridynamic model for diffusion. While similar, 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: in SG the discretization points (which have to be close to uniformly distributed throughout the domain, otherwise errors are large) have no geometrical dimensions, while in the meshfree PD discretization, nodes have an associated volume (area, length).

The spectral-based semi-analytical approach is limited to solving linear and homogeneous diffusion problems with insulated boundary conditions. Here we extended this procedure to Dirichlet boundary conditions and 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.

Acknowledgements

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

Reference

1. Ghosh, S., Das, N., Gonçalves, T., Quaresma, P., & Kundu, M. (2018). The journey of graph kernels through two decades. *Computer Science Review*, *27*, 88–111. https://doi.org/10.1016/j.cosrev.2017.11.002

2. Gasteiger, J., Engel, T., Gasteiger, J., Lekishvili, G., Engel, T., Kochev, N., … Engel, T. (2003). Chemoinformatics: A Textbook. https://doi.org/doi:10.1002/3527601643.fmatter

3. Xenarios, I., Salwński, L., Duan, X. J., 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 research*, *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.

5. Harchaoui, Z., & Bach, F. (2007). Image Classification with Segmentation Graph Kernels. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 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. *Advances in Natural Language Processing*, 324–336.

8. Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., & Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry, 34*(2), 786–797. https://doi.org/10.1021/jm00106a046

9. Kondor, R. I., & Lafferty, J. D. (2002). Diffusion Kernels on Graphs and Other Discrete Input

Spaces. *Proceedings of the Nineteenth International Conference on Machine Learning, 2002*, 315–22. Retrieved from http://portal.acm.org/citation.cfm?id=655996

10. Lafferty, J., & Lebanon, G. (2005). Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, *6*, 129–163. https://doi.org/10.1184/r1/6604859.v1

11. Kloster, K., & Gleich, D. F. (2014). Heat kernel based community detection. *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, K. D., & Rao, P. (2019). Thermal modeling in metal additive manufacturing using graph theory. *Journal of Manufacturing Science and Engineering*, *141*(7), 071007. https://doi.org/10.1115/1.4043648

13. Cole, K. D., Yavari, M. R., & Rao, P. K. (2020). Computational heat transfer with spectral graph theory : Quantitative verification. *International Journal of Thermal Sciences*, *153*, 106383. https://doi.org/10.1016/j.ijthermalsci.2020.106383

14. Silling, S. A. (2000). Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of 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. *International Journal of Fracture*, *196*(1–2), 59–98.

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. *Engineering Fracture Mechanics*, *219*, 106617. https://doi.org/10.1016/j.engfracmech.2019.106617

17. Hu, W., Ha, Y. D., & Bobaru, F. (2011). Modeling dynamic fracture and damage in a fiber-reinforced composite lamina with peridynamics. *International Journal for Multiscale Computational Engineering*, *9*(6), 707–726.

18. Hu, W., Ha, Y. D., & Bobaru, F. (2012). Peridynamic model for dynamic fracture in unidirectional fiber-reinforced composites. *Computer Methods in Applied Mechanics and Engineering*, *217*, 247–261.

19. Bobaru, F., & Duangpanya, M. (2010). The peridynamic formulation for transient heat conduction. *International Journal of Heat and Mass Transfer*, *53*(19–20), 4047–4059. https://doi.org/10.1016/j.ijheatmasstransfer.2010.05.024

20. Bobaru, F., & Duangpanya, M. (2012). A peridynamic formulation for transient heat conduction in bodies with evolving discontinuities. *Journal of Computational Physics*, *231*(7), 2764–2785. https://doi.org/10.1016/j.jcp.2011.12.017

21. Chen, Z., & Bobaru, F. (2015). Peridynamic modeling of pitting corrosion damage. *Journal of the Mechanics and Physics of Solids*, *78*, 352–381. https://doi.org/10.1016/j.jmps.2015.02.015

22. Jafarzadeh, S., Chen, Z., Zhao, J., & Bobaru, F. (2019). Pitting, lacy covers, and pit merger in stainless steel: 3D peridynamic models. *Corrosion Science*, *150*, 17–31. https://doi.org/10.1016/j.corsci.2019.01.006

23. Bobaru, F., Yang, M., Alves, L. F., Silling, S. A., Askari, E., & Xu, J. (2009). Convergence, adaptive refinement, and scaling in 1D peridynamics. *International Journal for Numerical Methods in Engineering*, *77*(6), 852–877. https://doi.org/10.1002/nme.2439

24. Silling, S. A., & Askari, E. (2005). A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures*, *83*(17–18), 1526–1535. https://doi.org/10.1016/j.compstruc.2004.11.026

25. Deo, N. (2017). *Graph theory with applications to engineering and computer science*. Courier Dover Publications.

26. Bıyıkoglu, T., Leydold, J., & Stadler, P. F. (2007). Laplacian eigenvectors of graphs. *Lecture notes in mathematics*, *1915*.

27. Merris, R. (1994). Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, *197*, 143–176.

28. Carslaw, H. S., & Jaeger, J. C. (1992). *Conduction of heat in solids*. Clarendon press.

29. Boyce, W. E., DiPrima, R. C., & Meade, D. B. (2017). *Elementary differential equations*. John Wiley & Sons.

30. Greub, W. H. (2012). *Linear algebra* (Vol. 23). Springer Science & Business Media.

31. Hammond, D. K., Gur, Y., & Johnson, C. R. (2013). Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. *2013 IEEE Global Conference on Signal and Information Processing*, 419–422. https://doi.org/10.1109/GlobalSIP.2013.6736904

32. 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 Networks*, *31*, 53–72. https://doi.org/10.1016/j.neunet.2012.03.001

33. Chen, Z., Niazi, S., & Bobaru, F. (2019). A peridynamic model for brittle damage and fracture in porous materials. *International Journal of Rock Mechanics and Mining Sciences*, *122*, 104059.

34. Jafarzadeh, S., Chen, Z., & Bobaru, F. (2018). Peridynamic modeling of intergranular corrosion damage. *Journal of The Electrochemical Society*, *165*(7), C362.

35. Zhao, J., Chen, Z., Mehrmashhadi, J., & Bobaru, F. (2018). Construction of a peridynamic model for transient advection-diffusion problems. *International Journal of Heat and Mass Transfer*, *126*, 1253–1266.

36. Chen, Z., & Bobaru, F. (2015). Selecting the kernel in a peridynamic formulation: A study for transient heat diffusion. *Computer Physics Communications*, *197*, 51–60.

37. Chen, Z., Ju, J. W., 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. *Engineering Fracture Mechanics*, *216*, 106498.

38. Chen, X., & Gunzburger, M. (2011). Continuous and discontinuous finite element methods for a peridynamics model of mechanics. *Computer Methods in Applied Mechanics and Engineering*, *200*(9–12), 1237–1250. https://doi.org/10.1016/j.cma.2010.10.014

39. Du, Q., Tian, L., & Zhao, X. (2013). A convergent adaptive finite element algorithm for nonlocal diffusion and peridynamic models. *SIAM Journal on Numerical Analysis*, *51*(2), 1211–1234. https://doi.org/10.1137/120871638

40. Jafarzadeh, S., Larios, A., & Bobaru, F. (2020). Efficient solutions for nonlocal diffusion problems via boundary-adapted spectral methods. *Journal of Peridynamics and Nonlocal Modeling*, 1–26.

41. Jafarzadeh, S., Wang., L., Larios, A., & Bobaru, F. (2021). A fast convolution-based method for peridynamic transient diffusion in arbitrary domains. *Comput. Methods Appl. Mech. Engrg*.

42. Murakami, S. (2012). *Continuum damage mechanics: a continuum mechanics approach to the analysis of damage and fracture* (Vol. 185). Springer Science & Business Media.

43. Silling, S. A., Epton, M., Weckner, O., Xu, J., & Askari, E. (2007). *Peridynamic states and constitutive modeling*. *Journal of Elasticity* (Vol. 88). https://doi.org/10.1007/s10659-007-9125-1

44. Wu, P., Zhao, J., Chen, Z., & Bobaru, F. (2020). Validation of a stochastically homogenized

peridynamic model for quasi-static fracture in concrete. *Engineering Fracture Mechanics*, *237*, 107293.

45. Bobaru, F., Ha, Y. D., & Hu, W. (2012). Damage progression from impact in layered glass modeled with peridynamics. *Open Engineering*, *2*(4), 551–561. https://doi.org/10.2478/s13531-012-0020-6

46. Le, Q. V, & Bobaru, F. (2018). Surface corrections for peridynamic models in elasticity and fracture. *Computational Mechanics*, *61*(4), 499–518.

47. Bobaru, F., & Ha, Y. D. (2011). Adaptive refinement and multiscale modeling in 2D peridynamics. *International Journal for Multiscale Computational Engineering*, *9*(6), 635–659.

48. Gu, X., Zhang, Q., & Xia, X. (2017). Voronoi-based peridynamics and cracking analysis with adaptive refinement. *International Journal for Numerical Methods in Engineering*, *112*(13), 2087–2109.

49. Henke, S. F., & Shanbhag, S. (2014). Mesh sensitivity in peridynamic simulations. *Computer Physics Communications*, *185*(1), 181–193.

50. Huilai, L., Zhuoqun, W., Jingxue, Y., & Junning, Z. (2001). *Nonlinear diffusion equations*. World Scientific.

51. Csanady, G. T. (2012). *Turbulent diffusion in the environment* (Vol. 3). Springer Science & Business Media.

52. Chen, Z., Jafarzadeh, S., Zhao, J., & Bobaru, F. (2021). A coupled mechano-chemical peridynamic model for pit-to-crack transition in stress-corrosion cracking. *Journal of the Mechanics and Physics of Solids*, *146*, 104203.

53. MATLAB. (2020). Natick, Massachusetts: The MathWorks Inc.

54. Higham, N. J. (2005). The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, *26*(4), 1179–1193.

55. Al-Mohy, A. H., & Higham, N. J. (2010). A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, *31*(3), 970–989.

56. Greenberg, M. D. (1988). *Advanced engineering mathematics*. Prentice-Hall.