International Journal on Artificial Intelligence Tools
Vol. 29, No. 5 (2020) 2050009 (12 pages)

DOI: 10.1142/S0218213020500098

Systematic Construction of Neural Forms for Solving Partial Differential Equations Inside Rectangular Domains, Subject to Initial, Boundary and Interface Conditions

Pola Lydia Lagari
School of Nuclear Engineering, Purdue University
West Lafayette, IN 47907, USA
plagari@purdue.edu

Lefteri H. Tsoukalas
School of Nuclear Engineering, Purdue University
West Lafayette, IN 47907, USA
tsoukala@purdue.edu

Salar Safarkhani
School of Mechanical Engineering, Purdue University
West Lafayette, IN 47907, USA
salar.safarkhani@gmail.com

Isaac E. Lagaris
Department of Computer Science and Engineering
University of Ioannina, Ioannina 45500, Greece
lagaris@cs.uoi.gr

Received 18 March 2020
Accepted 20 April 2020
Published 21 August 2020

A systematic approach is developed for constructing proper trial solutions to Partial Differential Equations (PDEs) of up to second order, using neural forms that satisfy prescribed initial, boundary and interface conditions. The spatial domain considered is of the rectangular hyper-box type. On each face either Dirichlet or Neumann conditions may apply. Robin conditions may be accommodated as well. Interface conditions that induce discontinuities, have not been treated to date in the relevant neural network literature. As an illustration a common problem of heat conduction through a system of two rods in thermal contact is considered.

Keywords: Interface conditions; neural forms; neural networks; partial differential equations.

1. Introduction

Partial Differential Equations (PDEs) find numerous applications in many scientific and engineering fields, therefore solution methods are of particular interest. Finite difference,[1] and finite element methods,[2] among others, have been considered in the past. Early Neural Network (NN) based methods[3–7] and more recently,[8–13] have attracted the interest of the scientific community, on the one hand, due to their effective approximation capability, and on the other hand, due to the analytic closed-form solution they offer.

In the present article, PDEs of second order in space will be considered. This category includes several important PDEs, such as

$$\text{Poisson equation:} \quad \nabla^2\Phi(x) = f(x) \tag{1a}$$

$$\text{Schrödinger equation:} \quad -\frac{\hbar^2}{2m}\nabla^2\Psi(x) + V(x)\Psi(x) = E\Psi(x) \tag{1b}$$

$$\text{Heat conduction equation:} \quad \frac{\partial}{\partial t}T(x,t) = \kappa\nabla^2 T(x,t) \tag{1c}$$

$$\text{Wave equation:} \quad \frac{\partial^2}{\partial t^2}u(x,t) = c^2\nabla^2 u(x,t) \tag{1d}$$

to mention just a few among others.

The solution domain is taken to be a rectangular hyper-box described by:

$$x \in \mathcal{B}^n \subset R^n \text{ with } \mathcal{B}^n = [a_1, b_1] \otimes [a_2, b_2] \otimes \cdots \otimes [a_n, b_n] \tag{2}$$

Given the PDE and the associated initial conditions (ICs) and/or boundary conditions (BCs), a trial solution is constructed using a “Neural Form”. A neural form (NF) is any expression that depends on a neural network. Neural forms may be used to impose certain functional properties. For example if by $N(x,\theta)$ we denote a NN with input $x \in R^n$, and weights collectively denoted by θ, then $F(x,\theta) = \|x - a\|N(x,\theta)$ is a NF, which vanishes at $x = a$. Similarly $F(x,\theta) = \|x\|^2 N(x,\theta)$ and its gradient, both vanish at $x = 0$.

The trial solution is cast as:

$$\Psi(x,t,\theta) = A(x,t) + Z(x,t)F(x,t,\theta) \tag{3}$$

where $F(x,t,\theta)$ is simply a NN if no interface conditions exist, otherwise it is a NF. Functions $A(x,t)$ and $Z(x,t)$ contain no adjustable parameters and $A(x,t)$ satisfies the problem's ICs and BCs, while $Z(x,t)$ is a function vanishing on the boundary, and only there, and does not contribute to the ICs and BCs. The trial solution is then substituted in the PDE and it is trained to satisfy it on a number of chosen points. This is accomplished in the lines of Refs. 3, 4, 6 and 7.

In Section 2, a systematic procedure is detailed for constructing functions $A(x,t)$ and $Z(x,t)$ for PDEs with orthogonal boundaries with either Dirichlet or Neumann boundary conditions on each side. In Section 3, a suitable NF is introduced to accommodate interface conditions required in a heat conduction problem, that is used to illustrate the approach, and in the last Section 4, a summary is given along with thoughts for future research and applications.

2. Boundary Matches

There are several types of boundary conditions:

- Dirichlet conditions, specify on the boundary, the value of the solution.
- Neumann conditions, specify on the boundary the value of the solution's normal derivative.
- Mixed conditions are those that on a part of the boundary are type Dirichlet, and on the remaining part are type Neumann.
- Robin conditions, specify on the boundary a linear combination of the solution's value and of its normal derivative.
- Cauchy conditions, specify on the boundary both the solution's value and its normal derivative.

Let us consider a function $f(t)$ with $t \geq t_0$. Initial conditions for first order in time equations, specify the value for $f(t)$ at $t = t_0$, and for second order equations, values for both $f(t)$ and its derivative at $t = t_0$. Boundary conditions for a function $f(x)$, with $x \in [a, b]$, specify values for $f(x)$ or for its derivative at two points $x = a$ and $x = b$. We seek to develop simple polynomial models that meet these conditions. In addition, when a or b or both are not finite, polynomials cannot model the boundary behavior since they diverge as $x \to \pm\infty$, so for these cases special models have been introduced.

2.1. *Single point matches and operators*

The quantity

$$P_{t_0}^n(t, f) \equiv \sum_{k=0}^{n} q_k(t_0, f) \frac{[1 - e^{-(t-t_0)}]^k}{k!}, \ \forall\, t \geq t_0 \tag{4a}$$

by choosing $q_k(t_0, f)$ appropriately, may satisfy the following requirements

$$\frac{d^k}{dt^k} P_{t_0}^n(t, f)_{|t=t_0} = f^{(k)}(t_0), \ \forall\, k = 0, 1, \ldots, n \tag{4b}$$

i.e. the kth derivative of $P_{t_0}^n(t, f)$ matches the kth derivative of $f(t)$ at $t = t_0$ for all $k = 0, 1, \ldots, n$. The first few coefficients are listed below

$$q_0(t_0, f) = f(t_0), \quad q_1(t_0, f) = f^{(1)}(t_0), \quad q_2(t_0, f) = f^{(2)}(t_0) + f^{(1)}(t_0) \tag{5}$$

We define the single point match-operator $\mathcal{I}_{t|t_0}^n$ by

$$\mathcal{I}_{t|t_0}^n f(t) \equiv P_{t_0}^n(t, f) \tag{6}$$

For an initial value problem, with $f(t_0)$ prescribed, a trial solution, that is finite for $t \to \infty$, may expressed by the following neural form

$$\Psi(t) = A(t) + Z(t)N(t, \theta) \tag{7a}$$

with

$$A(t) = \mathcal{I}^0_{t|t_0} f(t) = f(t_0) \tag{7b}$$

and

$$Z(t) = 1 - e^{-(t-t_0)} \tag{7c}$$

If both $f(t)$ and its derivative are prescribed at $t = t_0$ (Cauchy conditions), then a trial solution, that is finite for $t \to \infty$, may be given by

$$\Psi(t) = A(t) + Z(t)N(t, \theta) \tag{8a}$$

with

$$A(t) = \mathcal{I}^1_{t|t_0} f(t) = f(t_0) + (1 - e^{-(t-t_0)}) f^{(1)}(t_0) \tag{8b}$$

and

$$Z(t) = (1 - e^{-(t-t_0)})^2 \tag{8c}$$

2.2. *Two point matches and operators*

Let $\boldsymbol{P^{i,j}_{a,b}(x, f)}$ be a polynomial of minimal degree satisfying

$$\frac{\partial^i P^{i,j}_{a,b}}{\partial x^i}(x, f)|_{x=a} = f^{(i)}(a) \equiv \frac{\partial^i f}{\partial x^i}(x)|_{x=a} \tag{9a}$$

$$\frac{\partial^j P^{i,j}_{a,b}}{\partial x^j}(x, f)|_{x=b} = f^{(j)}(b) \equiv \frac{\partial^j f}{\partial x^j}(x)|_{x=b} \tag{9b}$$

The polynomial $\boldsymbol{P^{i,j}_{a,b}(x, f)}$, matches at the end-points (i.e. at the boundary) certain properties of the function $\boldsymbol{f(x)}$, and hence it may be termed to be a “boundary match”.

Let $\boldsymbol{\mathcal{L}^{i,j}_{x|a,b}}$ be an operator (the two-point match operator) defined as

$$\mathcal{L}^{i,j}_{x|a,b} f(x) \equiv P^{i,j}_{a,b}(x, f) \tag{10}$$

Then, the function

$$\boldsymbol{(1 - \mathcal{L}^{i,j}_{x|a,b}) f(x)}$$

has both vanishing the ith derivative at $x = a$ and the jth derivative at $x = b$.

For differential equations of second order in space, $i = 0, 1$ and $j = 0, 1$. The relevant two-point polynomial matches, are given explicitly by

$$P^{0,0}_{a,b}(x, f) = f(b)\frac{x-a}{b-a} - f(a)\frac{x-b}{b-a} \tag{11a}$$

$$P^{0,1}_{a,b}(x, f) = (x-a)f^{(1)}(b) + f(a) \tag{11b}$$

$$P_{a,b}^{1,0}(x,f) = (x-b)f^{(1)}(a) + f(b) \tag{11c}$$

$$P_{a,b}^{1,1}(x,f) = \frac{1}{2}\frac{f^{(1)}(b) - f^{(1)}(a)}{b-a}x^2 + \frac{bf^{(1)}(a) - af^{(1)}(b)}{b-a}x \tag{11d}$$

For a boundary value problem with prescribed $f(a)$ and $f^{(1)}(b)$, the following neural form may well serve for a trial solution

$$\Psi(x) = A(x) + Z(x)N(x,\theta) \tag{12a}$$

with

$$A(x) = \mathcal{L}_{x|a,b}^{0,1}f(x) = (x-a)f^{(1)}(b) + f(a) \tag{12b}$$

and

$$Z(x) = (x-a)(x-b)^2 \tag{12c}$$

When a is finite and $b \to \infty$, polynomials are not appropriate. In this case the match operators and the associated $Z(x)$ functions become

$$\mathcal{L}_{x|a,\infty}^{0,0}f(x) = f(a)e^{a-x} + f(\infty)(1-e^{a-x}) \tag{13a}$$

$$\mathcal{L}_{x|a,\infty}^{0,1}f(x) = f(a)e^{a-x} \tag{13b}$$

both with

$$Z(x) = (x-a)e^{-x} \tag{13c}$$

$$\mathcal{L}_{x|a,\infty}^{1,0} = -f^{(1)}(a)e^{a-x} + f(\infty) \tag{13d}$$

$$\mathcal{L}_{x|a,\infty}^{1,1} = -f^{(1)}(a)e^{a-x} \tag{13e}$$

both with

$$Z(x) = (x-a)^2e^{-x} \tag{13f}$$

Correspondingly, when b is finite and $a \to -\infty$ we have

$$\mathcal{L}_{x|-\infty,b}^{0,0}f(x) = f(-\infty)(1-e^{x-b}) + f(b)e^{x-b} \tag{14a}$$

$$\mathcal{L}_{x|-\infty,b}^{0,1}f(x) = f(-\infty) + f^{(1)}(b)e^{x-b} \tag{14b}$$

both with

$$Z(x) = (x-b)e^{x} \tag{14c}$$

$$\mathcal{L}_{x|-\infty,b}^{0,1} = f^{(1)}(b)e^{x-b} + f(-\infty) \tag{14d}$$

$$\mathcal{L}_{x|-\infty,b}^{1,1} = f^{(1)}(b)e^{x-b} \tag{14e}$$

both with

$$Z(x) = (x-b)^2e^{x} \tag{14f}$$

When $a \to -\infty$ and $b \to \infty$, then

$$\mathcal{L}^{0,0}_{x|-\infty,\infty} f(x) = f(\infty)\sigma(x) + f(-\infty)(1-\sigma(x)) \tag{15a}$$

$$\mathcal{L}^{1,0}_{x|-\infty,\infty} f(x) = f(\infty)\sigma(x) \tag{15b}$$

$$\mathcal{L}^{0,1}_{x|-\infty,\infty} f(x) = f(-\infty)(1-\sigma(x)) \tag{15c}$$

$$\mathcal{L}^{1,1}_{x|-\infty,\infty} f(x) = \sigma(x) \tag{15d}$$

all with

$$Z(x) = \sigma(x)(1-\sigma(x)) \tag{15e}$$

where

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{15f}$$

2.3. *Combining two-point matches, in two dimensions*

Consider a function of two variables: $f(x,y)$, $x \in [a,b]$, $y \in [c,d]$. The x-match operator acts (for example when $i = j = 0$) on $f(x,y)$ as

$$\mathcal{L}^{0,0}_{x|a,b}\, f(x,y) = f(b,y)\frac{x-a}{b-a} - f(a,y)\frac{x-b}{b-a}$$

The y-match operator acts (for example when $i = 1, j = 0$) on $f(x,y)$ as

$$\mathcal{L}^{1,0}_{y|c,d}\, f(x,y) = (y-d)\,\frac{\partial f}{\partial y}\,(x,c) + f(x,d)$$

The combined operation: (*Commutative*)

$$\mathcal{L}^{0,0}_{x|a,b}\,\mathcal{L}^{1,0}_{y|c,d}\, f(x,y) = \mathcal{L}^{1,0}_{y|c,d}\,\mathcal{L}^{0,0}_{x|a,b}\, f(x,y)$$

$$= \left[(y-d)\,\frac{\partial f}{\partial y}\,(b,c) + f(b,d)\right]\frac{x-a}{b-a} - \left[(y-d)\,\frac{\partial f}{\partial y}\,(a,c) + f(a,d)\right]\frac{x-b}{b-a}$$

2.4. *Building the boundary match in two dimensions*

The function

$$A(x,y) = \left[1 - \left(1 - \mathcal{L}^{i,j}_{x|a,b}\right)\left(1 - \mathcal{L}^{k,m}_{y|c,d}\right)\right] f(x,y)$$

$$= \left(\mathcal{L}^{i,j}_{x|a,b} + \mathcal{L}^{k,m}_{y|c,d} - \mathcal{L}^{i,j}_{x|a,b}\mathcal{L}^{k,m}_{y|c,d}\right) f(x,y)$$

matches (as it can be verified), the following BCs

$$\frac{\partial^i A}{\partial x^i}\,(a,y) = \frac{\partial^i f}{\partial x^i}\,(a,y), \quad \frac{\partial^j A}{\partial x^j}\,(b,y) = \frac{\partial^j f}{\partial x^j}\,(b,y)$$

$$\frac{\partial^k A}{\partial y^k}\,(x,c) = \frac{\partial^k f}{\partial y^k}\,(x,c), \quad \frac{\partial^m A}{\partial y^m}\,(x,d) = \frac{\partial^m f}{\partial y^m}\,(x,d)$$

Generalizing in Many Dimensions

Let, $x^T = (x_1, x_2, \ldots, x_N)$, $x_k \in [a_k, b_k]$, (rectangular boundary). Let, for x_k the BCs be represented by the match-operator $\mathcal{L}^{i_k, j_k}_{x_k|a_k, b_k}$. Then the associated multi-dimensional boundary match is given by

$$A(x) = \left[1 - \prod_{k=1}^{N} \left(1 - \mathcal{L}^{i_k, j_k}_{x_k|a_k, b_k}\right)\right] f(x)$$

and the corresponding appropriate Z-function is

$$Z(x) = \prod_{k=1}^{N} (x_k - a_k)^{1+i_k} (x_k - b_k)^{1+j_k}$$

2.5. *Robin boundary conditions*

The formalism may be extended to support also Robin BCs of the form

$$\hat{B}^L_i f(x) \equiv \left[\lambda^L_i f(x) + \mu^L_i \frac{\partial f(x)}{\partial x_i}\right]_{|x_i = a_i} \qquad \text{specified} \tag{16a}$$

$$\hat{B}^R_i f(x) \equiv \left[\lambda^R_i f(x) + \mu^R_i \frac{\partial f(x)}{\partial x_i}\right]_{|x_i = b_i} \qquad \text{specified} \tag{16b}$$

The operator $\hat{B}^L_i$ is the BC operator at the face $x_i = a_i$ (on the "Left") and $\hat{B}^R_i$ is the BC operator at the face $x_i = b_i$ (on the "Right").

Let the operator $\boldsymbol{\mathcal{M}_i}$ be defined as

$$\boldsymbol{\mathcal{M}_i} f(x) \equiv \Phi^R_i(x_i) \hat{B}^L_i f(x) - \Phi^L_i(x_i) \hat{B}^R_i f(x) \tag{17a}$$

with

$$\Phi^R_i(x_i) = \frac{\lambda^R_i (x_i - b_i) - \mu^R_i}{D_i}, \quad \Phi^L_i(x_i) = \frac{\lambda^L_i (x_i - a_i) - \mu^L_i}{D_i} \tag{17b}$$

$$D_i = (a_i - b_i)\lambda^L_i \lambda^R_i + \mu^L_i \lambda^R_i - \lambda^L_i \mu^R_i \tag{17c}$$

This expression satisfies the above mentioned Robin BCs. Note that the following relations hold

$$\hat{B}^R_i \Phi^R_i(x_i) = 0, \quad \hat{B}^L_i \Phi^L_i(x_i) = 0, \quad \hat{B}^L_i \Phi^R_i = 1, \quad \hat{B}^R_i \Phi^L_i = -1$$

Then the Robin-boundary match and the associated $Z(x)$ are given by

$$A(x) = \left[1 - \prod_{i=1}^{N} (1 - \mathcal{M}_i)\right] f(x), \quad Z(x) = \prod_{i=1}^{N} (x_i - a_i)^2 (x_i - b_i)^2$$

2.6. *Combining single and two-point operators*

Suppose that the function $f(x,t)$ with $x \in [a,b]$ and $t > t_0$, has prescribed two-point boundary conditions on x and single point initial conditions on t. If the relevant one and two-point operators are $\mathcal{I}^0_{t|t_0}$ and $\mathcal{L}^{i,j}_{x|a,b}$, the combined match operator is then given by

$$\mathcal{C}_0 = 1 - (1 - \mathcal{I}^0_{t|t_0})(1 - \mathcal{L}^{i,j}_{x|a,b}) = \mathcal{I}^0_{t|t_0} + \mathcal{L}^{i,j}_{x|a,b} - \mathcal{I}^0_{t|t_0}\mathcal{L}^{i,j}_{x|a,b} \tag{18}$$

If the single point operator is $\mathcal{I}^1_{t|t_0}$, that corresponds to Cauchy conditions, the combined operator is given by

$$\mathcal{C}_1 = \mathcal{I}^1_{t|t_0} + \mathcal{L}^{i,j}_{x|a,b} - \mathcal{I}^1_{t|t_0}\mathcal{L}^{i,j}_{x|a,b} \tag{19}$$

The neural forms for the trial solutions are given by

$$\Psi(x,t) = A(x,t) + Z(x,t)N(x,t,\theta), \quad \text{with } (\forall\, k = 0,1) \tag{20a}$$

$$A(x,t) = \mathcal{C}_k f(x,t) \tag{20b}$$

$$Z(x,t) = (x-a)^{1+i}(x-b)^{1+j}(1 - e^{-(t-t_0)})^{1+k} \tag{20c}$$

3. Interface Conditions: Heat-conduction PDE

For an illustrative example the focus is turned on the heat conduction equation, namely

$$\rho c^{(p)} \frac{\partial}{\partial t} T(x,t) = k \frac{\partial^2}{\partial x^2} T(x,t), \text{ with } \; x \in [a,c], \; t \ge t_0 \tag{21}$$

subject to

$$\text{IC:} \quad T(x,t_0) \qquad \text{prescribed} \tag{22a}$$

$$\text{BCs } (\forall\, t > t_0): \quad T(a,t) \text{ or } \frac{\partial}{\partial x} T(a,t) \qquad \text{prescribed} \tag{22b}$$

$$T(c,t) \text{ or } \frac{\partial}{\partial x} T(c,t) \qquad \text{prescribed} \tag{22c}$$

The trial solution for Eq. (21), subject to (22), may be modeled as

$$\Psi(x,t,\theta) = A(x,t) + Z(x,t)N(x,t,\theta) \tag{23}$$

The trial solution in Eq. (23), assumes that the thermal conductivity k, is constant for $x \in [a,c]$, in which case both $\Psi(x,t,\theta)$ and $\frac{\partial}{\partial x}\Psi(x,t,\theta)$ are continuous functions of x. However when conduction takes place through two different materials that are in contact, this is no longer true.

The situation of interest is depicted in Fig. 1, where two rods are laid side by side along the x-axis, and are in contact at $x = b$. Each rod is characterized by its density ρ_i, specific heat $c_i^{(p)}$, and thermal conductivity k_i, where the subscript $i \in \{1,2\}$ labels correspondingly, quantities related to the rod on the left and to

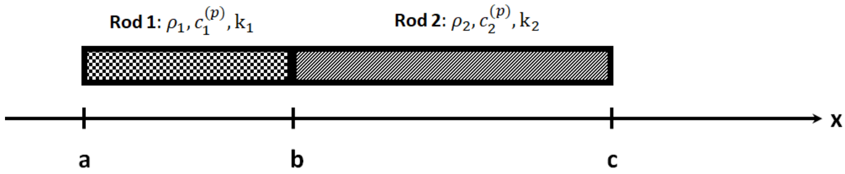

Fig. 1. Two rods placed side by side along the x-axis, with density, specific heat and thermal conductivity values, ρ_i, $c_i^{(p)}$, and k_i.

the rod on the right. At the interface (point $x = b$) the following conditions must be met

$$\lim_{\epsilon \to 0^+} T(b - \epsilon, t) = \lim_{\epsilon \to 0^+} T(b + \epsilon, t) \tag{24a}$$

$$\lim_{\epsilon \to 0^+} k_1 \frac{\partial}{\partial x} T(b - \epsilon, t) = \lim_{\epsilon \to 0^+} k_2 \frac{\partial}{\partial x} T(b + \epsilon, t) \tag{24b}$$

Clearly Eq. (24b) introduces a discontinuity at $x = b$ for the temperature gradient. Therefore the model in (23) needs to be modified to handle condition (24b). This discontinuity will be introduced by replacing the NN in (23) with a NF as

$$\Psi(x, t, \theta) = A(x, t) + Z(x, t)[N(x, t, \theta) + \lambda(t, \theta)|x - b|] \tag{25}$$

where $\lambda(t, \theta)$ is a function to be determined so that condition (24b) is satisfied. Consider that the temperatures at the two end points, $T(a, t)$ and $T(c, t)$ for $t > t_0$ are prescribed. Then applying the corresponding match operator

$$\begin{aligned} A(x, t) &= \left[\mathcal{I}^{0}_{t|t_0} + \mathcal{L}^{0,0}_{x|a,c} - \mathcal{I}^{0}_{t|t_0}\mathcal{L}^{0,0}_{x|a,c}\right] T(x, t) \\ &= T(x, t_0) + \left[T(c, t)\frac{x - a}{c - a} - T(a, t)\frac{x - c}{c - a}\right] \\ &\quad - \left[T(c, t_0)\frac{x - a}{c - a} - T(a, t_0)\frac{x - c}{c - a}\right] \end{aligned} \tag{26a}$$

$$Z(x, t) = (x - a)(x - c)(1 - e^{-(t - t_0)}) \tag{26b}$$

Since $T(x, t_0)$ is the solution at time $t = t_0$, it satisfies both the interface conditions (24). Requiring that $\Psi(x, t > t_0, \theta)$ satisfies them as well, we arrive at the following expression for $\lambda(t, \theta)$, for $t > t_0$.

$$\begin{aligned} \lambda(t, \theta) = \frac{k_1 - k_2}{k_1 + k_2}\Bigg[&\frac{\partial N(b, t, \theta)}{\partial x} + \frac{2b - a - c}{(b - a)(b - c)} N(b, t, \theta) \\ &+ \frac{[T(c, t) - T(c, t_0)] - [T(a, t) - T(a, t_0)]}{(c - a)(b - a)(b - c)(1 - e^{-(t - t_0)})}\Bigg] \end{aligned} \tag{27}$$

The error function to be optimized is then given by:

$$E(\theta) = \sum_{t_i>0} \sum_{x_j \in (a,b)} \left[\frac{\partial \Psi(x_j,t_i,\theta)}{\partial t} - \frac{k_1}{\rho_1 c_1^{(p)}} \frac{\partial^2 \Psi(x_j,t_i,\theta)}{\partial x^2} \right]^2 + \sum_{t_i>0} \sum_{x_j \in (b,c)} \left[\frac{\partial \Psi(x_j,t_i,\theta)}{\partial t} - \frac{k_2}{\rho_2 c_2^{(p)}} \frac{\partial^2 \Psi(x_j,t_i,\theta)}{\partial x^2} \right]^2 \tag{28}$$

3.1. *Numerical example and results*

A specific example was solved corresponding to a rod of aluminum in contact to a rod of steel. The aluminum rod extends from $a = 0$ cm to $b = 5$ cm and the steel rod from $b = 5$ cm to $c = 12$ cm. The numerical values of the material parameters are depicted in Table 1 and the initial, boundary and interface conditions in Table 2.

Table 1. Rod properties.

Material	k, in $W/(\mathrm{cm} \cdot \mathrm{K})$	ρ, in g/cm^3	$c^{(p)}$, in J/(g $\cdot$ K)
Aluminum	$k_1 = 2.0$	$\rho_1 = 2.70$	$c_1^{(p)} = 0.92$
Steel	$k_2 = 0.5$	$\rho_2 = 7.85$	$c_2^{(p)} = 0.51$

Table 2. Initial, boundary and interface conditions ($\epsilon \to 0^+$).

$T(x,0) = 10°\mathrm{C}$	
$T(0,t>0) = 5°\mathrm{C}$	$T(12,t>0) = 15°\mathrm{C}$
$T(5-\epsilon,t) = T(5+\epsilon,t)$	$k_1 \dfrac{\partial T(5-\epsilon,t)}{\partial x} = k_2 \dfrac{\partial T(5+\epsilon,t)}{\partial x}$

The neural network employed was a perceptron with one hidden layer and sigmoid activation functions, namely

$$N(x,t,\theta) = \sum_{i=1}^{nodes} \theta_{4i-3}\sigma(\theta_{4i-2}x + \theta_{4i-1}t + \theta_{4i}) \tag{29}$$

where the sigmoid activation $\sigma(x)$, is given in formula (15f).

The x-grid used 120 points, while the t-grid used a step of $\tau = 0.05$ s, and the equation was solved up to $t = 30$ s (i.e. 600 steps). A plot of $T(x,t)$ for $t = 1$ s, 4 s, and 30 s is displayed in Fig. 2. Note that the solution was obtained with only five nodes in the hidden layer ($nodes = 5$). At $t \to \infty$, the solution is easily recovered by setting the time derivative to zero. Hence, $T(0 \le x < 5, \infty) = \frac{10}{33}x + 5$, and $T(5 < x \le 12, \infty) = \frac{40}{33}x + \frac{15}{33}$.

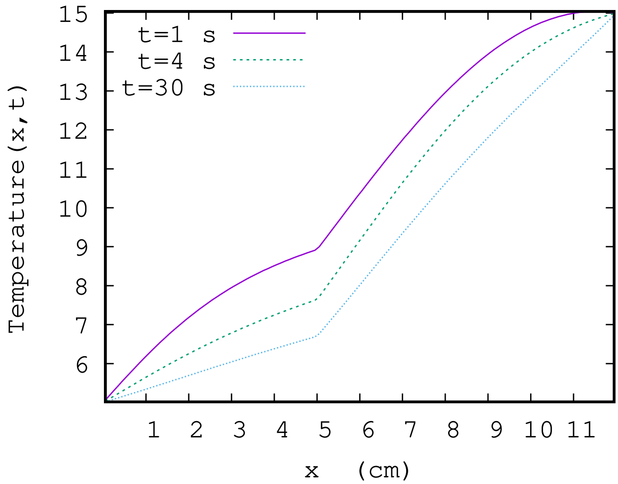

Fig. 2. Plots of the temperature spatial distribution at three different times ($t = 1$ s, 4 s and 30 s).

4. Conclusions

Given a partial differential equation of second order in space and up to second order in time, defined inside rectangular hyperboxes, we have developed a systematic approach for constructing a trial solution as: $\Psi(x,t,\theta) = A(x,t) + Z(x,t)F(x,t,\theta)$. $A(x,t)$ is a function without adjustable parameters satisfying the initial and boundary conditions. $Z(x,t)$ is vanishing on the boundary and such that it does not contribute to the initial and boundary conditions. $F(x,t,\theta)$ is a neural form when interface conditions exist, or simply a neural network otherwise.

We have demonstrated the applicability of the method by solving a problem with interface conditions, a case which, to the best of our knowledge, has not been tackled before, in the framework of the neural network methodologies for differential equations. It will be interesting to apply this methodology to problems with various interface geometries and extend it for the case of moving boundary problems, that are considered to be very hard.

Acknowledgments

This work has been supported by a donation to AI Systems Lab (AISL) by GS Gives, the Consortium for Nonproliferation Enabling Capabilities (CNEC) and the Office of Naval Research (ONR) under Grant No. N00014-18-1-2278.

References

1. G. D. Smith, *Numerical Solution of Partial Differential Equations. Finite Difference Methods* (Oxford University Press, 1971).
2. P. Šolin, *Partial Differential Equations and the Finite Element Method*, 1st edn. (Wiley-Interscience, 2006).

3. I. E. Lagaris, A. Likas and D. I. Fotiadis, Artificial neural network methods in quantum mechanics, *Comput. Phys. Commun.* **104** (1997) 1–14.
4. I. E. Lagaris, A. Likas and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* **9** (1998) 987–1000.
5. I. E. Lagaris, A. Likas and D. G. Papageorgiou, Neural network methods for boundary value problems with irregular boundaries, *IEEE Transactions on Neural Networks* **11** (2000) 1041–1049.
6. L. P. Aarts and P. van der Veer, Neural network method for solving partial differential equations, *Neural Processing Letters* **14**(3) (2001) 261–271.
7. K. S. McFall and J. R. Mahan, Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions, *Trans. Neur. Netw.* **20** (August 2009) 1221–1233.
8. S. Mall and S. Chakraverty, Single layer Chebychev neural network model for solving elliptic differential equations, *Neural Processing Letters* **45** (2017) 825–840.
9. J. Berg and K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, *Neurocomputing* **317** (2018) 28–41.
10. J. Sirignano and K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *Journal of Computational Physics* **375** (2018) 1339–1364.
11. M. Raissi, P. Perdikaris and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* **378** (2019) 686–707.
12. A. Namadchian and M. Ramezani, Analytical solution of stochastic differential equation by multilayer perceptron neural network approximation of Fokker–Planck equation, *Numerical Methods for Partial Differential Equations* **36**(3) (2020) 637–653.
13. C. Leake and D. Mortari, Deep theory of functional connections: A new method for estimating the solutions of partial differential equations, *Machine Learning and Knowledge Extraction* **2**(1) (2020) 37–55.