Applied Soft Computing 12 (2012) 3051–3062

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Solving differential equations with Fourier series and Evolution Strategies

Jose M. Chaquet*, Enrique J. Carmona

Dpto. de Inteligencia Artificial, Escuela Técnica Superior de Ingeniería Informática, Universidad Nacional de Educación a Distancia, Madrid, Spain

ARTICLE INFO

Article history:
Received 4 November 2011
Received in revised form 11 April 2012
Accepted 8 May 2012
Available online 30 May 2012

Keywords:
Differential equations
Fourier series
Evolution Strategies
Mesh-free methods
Harmonic analysis

ABSTRACT

A novel mesh-free approach for solving differential equations based on Evolution Strategies (ESs) is presented. Any structure is assumed in the equations making the process general and suitable for linear and nonlinear ordinary and partial differential equations (ODEs and PDEs), as well as systems of ordinary differential equations (SODEs). Candidate solutions are expressed as partial sums of Fourier series. Taking advantage of the decreasing absolute value of the harmonic coefficients with the harmonic order, several ES steps are performed. Harmonic coefficients are taken into account one by one starting with the lower order ones. Experimental results are reported on several problems extracted from the literature to illustrate the potential of the proposed approach. Two cases (an initial value problem and a boundary condition problem) have been solved using numerical methods and a quantitative comparative is performed. In terms of accuracy and storing requirements the proposed approach outperforms the numerical algorithm.

1. Introduction

Differential equations are mathematical equations for one or several unknown functions that relate the values of the functions themselves and their derivatives of various orders. Differential equations play a prominent role in engineering, physics, economics, and other disciplines. Some important examples are the Newton's Second Law in dynamics, the Maxwell's equations in electromagnetism, the heat equation in thermodynamics, Einstein's field equation in general relativity, Schrödinger equation in quantum mechanics or the Navier–Stokes equations in fluid dynamics [6].

Some simple differential equations admit solutions given by explicit formulas. But in the general case, only approximate solutions can be found. Among the engineering community, the most popular methods for solving differential equations use numerical analysis techniques such as finite element method (FEM) [21], finite difference method [14], or finite volume method [11]. These approaches relied on a grid or a mesh for discretizing the equations, bringing them into a finite-dimensional subspace. The original problem is reduced to the solution of algebraic equations.

On the other hand, mesh-free methods work with a set of arbitrary distributed points without using any mesh that provides the connectivity of these nodes. Some examples of mesh-free methods are smoothed particle hydrodynamics (SPH), diffusive element method (DEM) and point interpolation method (PIM) among others [12].

Other mesh-free methods have its inspiration in the artificial intelligence field. For instance Lagaris et al. [10] use a feed forward neural network to codify the solution of this type of problems. The trial solutions are computed as a sum of two parts. The first part satisfies the initial/boundary conditions and contains no adjustable parameters. The second part is constructed so as not to affect the initial/boundary conditions. This approach is problem dependent and in some cases could be difficult to split the candidate solutions in the two terms. Neural network weights and bias are optimized using a quasi-Newton Broyden–Fletcher–Goldfarb–Shanno method. This approach has been successfully applied to a system of partial differential equations which models a non-steady fixed bed non-catalytic solid-gas reactor [15]. In this last work, the boundary condition error is added to the cost or fitness function as a penalty term.

Nowadays an increasing interest in solving differential equations using artificial neural networks is observed. In [19] a Multilayer Perceptron and radial basis function neural network are successfully applied to the nonlinear Schröndinger equation in hydrogen atom. Yazdi and Pourreza [24] combine a neural network and a fuzzy system to solve some simple first and second order ordinary linear differential equations. Fast convergence is achieved training the adaptive network-based fuzzy inference system in unsupervised way. In [3] other mesh-free numerical method for solving PDEs based on integrated radial basis function networks with adaptive residual subsampling training scheme is presented. Numerical experiments solving several PDEs show that this algorithm with the adaptive procedure requires fewer neurons to attain

* Corresponding author. Tel.: +34 91 398 7301.
E-mail addresses: jose.chaquet@gmail.com (J.M. Chaquet), ecarmona@dia.uned.es (E.J. Carmona).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.05.014

the desired accuracy than conventional radial basis function networks. A different approach dealing with neural networks consist of solving a family of differential equations using traditional methods and training a neural network for building surrogate models. Following this line, work [5] presents a new hybrid adaptive neural network with modified adaptive smoothing errors based on genetic algorithm to construct a learning system for complex problem solving in fluid dynamics. The system can predict an incompressible viscous fluid flow represented by a stream function through symmetrical backward-facing steps channels.

Recently new methods for solving differential equations using Genetic Programming (GP) have been reported. These approaches can be considered mesh-free methods because the derivatives are computed symbolically, so any node connectivity is needed. Sobester et al. [20] propose a technique for the mesh-free solution of elliptic partial differential equations where least-squares collocation principle has been employed to define an appropriate objective function, which is optimized using GP. In that work no particular function basis is used, but symbolic regression is performed. This makes the search space very large. Another GP approach can be seen in [8] where polynomials are used for solving the convective-diffusion equation. In the same line of research, Tsoulos and Lagaris [23] use a similar technique, with the novelty of evolving the candidate solutions using Grammatical Evolution (GE) [13]. This technique has been employed successfully for solving the matrix Ricatti differential equation for nonlinear singular system [1]. GE has been used as well for enhanced the constructed neural method in [22]. The main advantage of this last approach is that the user does not choose a priori the number of neuron cells. In that contribution local search is employed over some individuals.

Seaton et al. [18] investigate the influence of the problem complexity and perform a search analysis when differential equations are solved within an evolutionary framework. They show that reducing the search space can improve significantly the algorithm performances. A possible approach for reducing the search space dimension is using some kind of function basis for building candidate solutions. This idea is used by Kirstukas et al. [9], where a hybrid GP approach from an engineering perspective is employed. In that approach, for the particular case of linear differential equations, a modified Gram–Schmidt algorithm is used to reduce the set of general solutions located by GP to a function basis set.

In the present work a novel mesh-free method for solving differential equations is reported. Candidate solutions are expressed as partial sums of Fourier series. In order to simplify the problem, an even periodic expansion of the solutions is done in such a way that all the sine coefficients are vanished. This representation can be regarded equivalent to a Discrete Cosine Transform (DCT) [17] which has been successfully used in several science and engineering applications, as for lossy compression of audio (MP3) and image (JPEG). With the chosen solution representation, the problem of solving differential equations is transformed into an optimization one, where the differential equation residuals and the boundary condition errors are minimized. The optimal Fourier coefficients are sought using Evolution Strategies (ESs). In order to systematize the process, the harmonic searching is done in a progressive way starting with the lowest order harmonic and using a different ES cycle to find the optimum value for each one.

The rest of the paper is organized as follows: In Section 2 a description of the proposed approach is given. In Section 3 a set of test cases extracted from the literature is described and experimental results are reported. Section 4 gives some qualitative and quantitative comparisons with numerical methods and other evolutionary approaches. Finally, the conclusions and some future work guides are outlined in Section 5.

2. Method description

In this section the proposed method is described. First the mathematical statement of the problem is given in Section 2.1. The particular coding of candidate solutions using Fourier series is explained in Section 2.2. Each optimal harmonic coefficient is sought using several ES steps. Section 2.3 describes these particular steps, and Section 2.4 explains how the steps are combined for solving the global optimization problem.

2.1. Statement of the problem

Using the same notation than Sobester et al. [20] but extending the original problem to systems of differential equations, we consider the general equation

$$\mathbf{Ly}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \quad \text{in} \ \Omega \subset \mathbb{R}^d \tag{1}$$

subject to the boundary conditions

$$\mathbf{By}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \quad \text{on} \ \partial\Omega, \tag{2}$$

where **L** and **B** are differential operators in the space $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y}(\mathbf{x})$ denotes the unknown solution vector. Functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ denote source terms, so only depend on **x**, but not on **y** or its derivatives. From a general point of view, $\mathbf{y}(\mathbf{x})$, $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ belong to the set of vector-valued functions $\mathbb{R}^d \to \mathbb{R}^m$. $\Omega \subset \mathbb{R}^d$ is a bounded domain and $\partial\Omega$ denotes its boundary.[1] Note that if $d = 1$ and $m = 1$, we have an ODE problem. If $d = 1$ and $m > 1$, a SODE problem is managed and, finally, if $d > 1$ and $m = 1$, a PDE problem is established. The solution vector satisfying (1) and (2) can be computed solving the following *Constrained Optimization Problem* (COP):

$$\begin{aligned} &\text{Minimize}: \int_{\Omega} \|\mathbf{Ly}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|^2 \, d\mathbf{x} \\ &\text{Subject to:} \int_{\partial\Omega} \|\mathbf{By}(\mathbf{x}) - \mathbf{g}(\mathbf{x})\|^2 \, d\mathbf{x} = 0 \end{aligned} \tag{3}$$

where $||\cdot||$ denotes the Euclidean norm in $\mathbb{R}^d$ space. This problem is discretized using a set of collocation points $C = \{(\mathbf{x}_i)|_{i=1,\ldots,n_C} \subset \Omega\}$ situated within the domain and as well on the boundary $B = \{(\mathbf{x}_j)|_{j=1,\ldots,n_B} \subset \partial\Omega\}$. Finally the original COP is transformed into a *Free Constrained Optimization Problem* defining a cost function as follows

$$F(\mathbf{y}) = \frac{1}{d\cdot(n_C + n_B)}\left[\sum_{i=1}^{n_C}\|\mathbf{Ly}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_i)\|^2 + \varphi\sum_{j=1}^{n_B}\|\mathbf{By}(\mathbf{x}_j) - \mathbf{g}(\mathbf{x}_j)\|^2\right], \tag{4}$$

where φ is a penalty parameter. Note that the cost function is obtained dividing the residuals by the total number of collocation points $d\cdot(n_C + n_B)$ in a similar way than Parisi et al. [15]. Other authors [1,10,20] do not make this normalization, which makes their values more dependent on the number of collocation points.

2.2. Candidate solutions

In the proposed approach, each component $y(\mathbf{x})$ of the trial solution is expressed as a partial sum of a Fourier series. The periodic expansion of $y(\mathbf{x})$ from the original definition range to all $\mathbb{R}^d$ is always performed using even functions. Therefore all the sine Fourier coefficients are vanished. In order to define this expansion, first some notation must be introduced. For each coordinate x_k with $k = 1, \ldots, d$, variables $x_{k,min}$ and $x_{k,max}$ are defined as the minimum

[1] This notation corresponds to elliptic equations appearing in the solution of boundary value problems. Other kind of differential equations such as initial value problems can be treated in a similar way.