

Chapter 14

DQ Based Methods: Theory and Application to Engineering and Physical Sciences

Stefania Tomasiello
University of Basilicata, Italy

ABSTRACT

Though relatively unknown, the Differential Quadrature Method (DQM) is a promising numerical technique that produces accurate solutions with less computational effort than other numerical methods such as the Finite Element Method. There are different versions of the former method, so one can refer to a class of methods based on the differential quadrature (DQ) approach. This chapter provides systematic steps to understand DQ based methods: how they work, how to use and develop them, as well as exemplary problems. It is divided into two sections: the first provides fundamentals and theories related to the DQ approach, while the second presents the application of DQ based methods to three significant problems, i.e. free vibrations of a multi-span beam (a typical model for bridge decks and floors), time dependent heat transfer, and vibrations of a rectangular membrane. The main focus is on the application of the method to space-time domains as a whole, an issue not well covered by the current literature.

INTRODUCTION

A variety of numerical methods are available for solving problems in science and engineering, when analytical solutions (so-called closed-form solutions) cannot be achieved. Numerical methods are part of the field called *scientific computing*, which can be defined as (Golub and Ortega, 1992, p. 2):

“The collection of tools, techniques, and theories required to solve on a computer mathematical models of problems in science and engineering”.

Mathematical models very often take the form of differential equations, with initial and/or boundary conditions. The variables of the problem may depend on time and/or n -space coordinates respectively (in this latter case the problem is called n -dimensional, where $n=1,2,3$ usually). For example, a boundary condition can represent

DOI: 10.4018/978-1-61350-116-0.ch014

restraints i.e. the supports of the multi-span beam (see the first example in the final section) and an initial condition can be the initial heat profile in the time-dependent problem of heat transfer (see the second example in the final section).

Numerical computations are affected by several types of error, *rounding errors* are due to machine precision. *Discretization errors* are created when the “continuous” problem domain is substituted by a “discrete” one. The *convergence error* is due to the iterative nature of some numerical methods, where a finite sequence of approximations to a solution is generated. An acceptable degree of error depends on the problem being solved: for many problems in industry or engineering three digit accuracy is quite acceptable.

Discretization is the core of almost all the numerical methods (including the methods discussed in this chapter), by introducing a set of points into the problem or computational spatial domain (i.e. where the solution will be computed). These points, often equally spaced, can have different names, e.g. *grid points*, in the Finite Difference Method (FDM), *collocation points* in the collocation methods or *nodes* in the Finite Element Method (FEM). Choosing the points in the computational domain is achieved by constructing the approximate solution of the values at these points. This has the benefit of converting the differential equations into an algebraic system of equations that can be easily solved on a computer.

Replacing the differential equations with an algebraic system of equations can be done by approximating the derivatives of the unknown function (the solution of the problem) by formulas based on the values of the function itself at discrete points and is generally achieved in a systematic way (e.g. in FDM) or by using polynomials which interpolate the function at the given points (e.g. FEM or collocation methods). How these interpolating polynomials are chosen and used is the main difference between some numerical methods. In collocation methods a finite-dimensional space of candidate solutions (usually, polynomials up

to a certain degree) are chosen, so that the solution satisfies the given equation at the collocation points. The FEM approximates the solution as a linear combination of piecewise functions (usually, polynomials), that are nonzero on small sub-domains. In fact, the basic idea of FEM is to divide the spatial domain of the problem into smaller parts (or sub-domains) called *finite elements* or *elements*, connected by points called *nodes*, giving a topological map which is called a *mesh*, whereas the process of making the mesh is called *mesh generation*. The main advantage of the FEM is its ability to reproduce non-smooth solutions, in conjunction with the ability to handle arbitrarily shaped domains. For example, stress analysis is an important process in engineering that requires the solution of a system of partial differential equations that are very difficult to solve by analytical methods except for very simple shapes, such as rectangles; however, engineering problems seldom involve such simple shapes. For linear problems, the solution is determined by solving a system of linear equations, for a certain number of unknowns, which are the nodal values of the problem function. With regard to engineering problems, in order to obtain a reasonably accurate solution, thousands of nodes are usually needed. Generally, the accuracy of the solution improves as the number of elements (and nodes) increases, but the computing time (and hence the cost) also rises.

For numerical methods, as well the question of accuracy, one also has to consider that of *efficiency*, which can be defined as the effort (both human and computer) required for solving a given problem. Generally a “computationally efficient method” is a method that minimizes computing time and retains suitable accuracy in the “approximate” solution. It is a reasonable research aim to find techniques that combine accuracy with relatively small computational costs (i.e. roughly speaking, a small number of discrete points).

In recent years, the Differential Quadrature Method (DQM) has become an increasingly

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/based-methods-theory-application-engineering/60366

Related Content

Cybersecurity and Data Breaches at Schools

Libi Shen, Irene Chen and Anchi Su (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 1294-1317).

www.irma-international.org/chapter/cybersecurity-and-data-breaches-at-schools/203561

A Test-Driven Approach to Behavioral Queries for Service Selection

Laura Zavala, Benito Mendoza and Michael N. Huhns (2013). *Agile and Lean Service-Oriented Development: Foundations, Theory, and Practice* (pp. 116-133).

www.irma-international.org/chapter/test-driven-approach-behavioral-queries/70732

Design and Optimizations of Lattice Boltzmann Methods for Massively Parallel GPU-Based Clusters

Enrico Calore, Alessandro Gabbana, Sebastiano Fabio Schifano and Raffaele Tripiccione (2018). *Analysis and Applications of Lattice Boltzmann Simulations* (pp. 54-114).

www.irma-international.org/chapter/design-and-optimizations-of-lattice-boltzmann-methods-for-massively-parallel-gpu-based-clusters/203087

Review of Applications of Energy Harvesting for Autonomous Wireless Sensor Nodes

Wilma Pavitra Puthran, Sahana Prasad and Rathishchandra Ramachandra Gatti (2023). *Energy Systems Design for Low-Power Computing* (pp. 143-165).

www.irma-international.org/chapter/review-of-applications-of-energy-harvesting-for-autonomous-wireless-sensor-nodes/319994

Learning With Software-Defined Area

Anurag Tiwari and Suneet Gupta (2018). *Innovations in Software-Defined Networking and Network Functions Virtualization* (pp. 291-305).

www.irma-international.org/chapter/learning-with-software-defined-area/198204