SPRINGER BRIEFS IN COMPUTER SCIENCE

Roman Trobec
Gregor Kosec

Parallel Scientific Computing

Theory, Algorithms, and Applications of Mesh Based and Meshless Methods

Springer

SpringerBriefs in Computer Science

More information about this series at http://www.springer.com/series/10028

Roman Trobec · Gregor Kosec

Parallel Scientific Computing

Theory, Algorithms, and Applications of Mesh Based and Meshless Methods

Springer

Roman Trobec
Jožef Stefan Institute
Ljubljana
Slovenia

Gregor Kosec
Jožef Stefan Institute
Ljubljana
Slovenia

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-17072-5 ISBN 978-3-319-17073-2 (eBook)
DOI 10.1007/978-3-319-17073-2

Library of Congress Control Number: 2015934954

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

To all who make our lives worthwhile.

Preface

The scientific computing and computer simulations on modern, powerful computers are tools that can reduce the costs of developing new machines, evaluate different environmental risks, simulate the evolution of different natural or technological phenomena, and conduct virtual experiments that are too dangerous or impossible to perform in laboratories, amongst many other possibilities.

This book is concentrated on the synergy between computer science and numerical analysis. It is written to provide a firm understanding of the described approaches to computer scientists, engineers or other experts who have to solve real problems. The meshless solution approach is described in more detail, with a description of the required algorithms and the methods that are needed for the design of an efficient computer program. Most of the details are demonstrated on solutions of practical problems, from basic to more complicated ones. We believe that this book will be a useful tool for any reader interested in solving complex problems in real computational domains.

We are grateful to all our colleagues who have contributed to this book through discussions or by reading the material, in particular to Marjan Šterk and Božidar Šarler who initiated and supported the research on meshless methods in our research community. Many thanks to Monika Kapus-Kolar and Matjaž Depolli, who carefully read our text and resolved many formal and linguistic inconsistencies. We are indebted to the Jožef Stefan Institute and the Slovenian Research Agency for their support of our work.

Ljubljana, February 2015

Roman Trobec
Gregor Kosec

Contents

Chapter 1
Introduction

Abstract The background and motivation for the development of solution methodologies for partial differential equations are given with an overview of the related work and the relevant publications.

Keywords Modeling · Partial differential equation · Numerical solution

1.1 Overview and Motivation

In the recent decades, scientific computing and numerical modeling have been drawing a lot of attention in research, due to their major contribution to a better understanding of Nature as well as in the development of advanced technologies. The modeling of more and more complex physical transport systems helps the community to address important issues like identifying environmental problems, improving technological processes, developing biomedical applications, etc. Many physical models are constituted through systems of coupled Partial Differential Equations (PDEs). Unfortunately, most of the PDEs that describe real-life problems do not possess a closed-form solution and, therefore, a suitable numerical approach is required. In the majority of numerical simulations, the Finite Volume Method (FVM), the Finite Difference Method (FDM), the Boundary Element Method (BEM), or the Finite Element Method (FEM) are used. However, there are also numerous scientific works related to the development and implementation of a relatively new class of simulation methods, referred to as meshless or meshfree methods.

Regardless of the numerical method, the solution algorithms are executed on computers, and so in most cases the accuracy of the computed solution is limited by the capacities of the available computer resources and by the efficiency of the computer programs implementation. Numerical modeling and computer science are, therefore, closely coupled scientific disciplines. The developments of computer technology are nowadays extremely vivid. Almost all modern computer platforms are parallel; most computers use several computing cores, sharing the same memory. For more complex computations, interconnected computers in computing clusters that work with a distributed memory are used. Moreover, the use of Graphical Processing Units

R. Trobec and G. Kosec, *Parallel Scientific Computing*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-17073-2_1

(GPUs) and Field Programmable Gate Arrays (FPGAs) for accelerating numerical simulations are becoming more and more attractive.

To effectively analyze complex natural phenomena by means of numerical simulations, all the involved phases, i.e., the physical modeling, mathematical formulation of the numerical methods, and their computer implementation and execution, have to be adequately addressed. This book tackles, in limited detail, all three phases, where we focus our discussion on the phenomena that can be described with a coupled system of second-order PDEs and supporting constitutive equations. We are, in particular, interested in the computational aspects of local meshless numerical methods, which, unlike the usual mesh-based methods, like the FEM or the FDM, require no topological relations between the discretization nodes.

Two different classes of local meshless methods are analyzed, i.e., strong form and weak form meshless methods. Regardless of the form of the equations, the basic principle of the local meshless methods is to create a local approximation of the solution that can be further manipulated, in most cases, with partial differential operators. The Meshless Local Strong Form Method (MLSM) is a generalization of the methods in the literature known as the Diffuse Approximate Method (DAM) [1], Local Radial Basis Function Collocation Method (LRBFCM) [2], Generalized FDM [3], Collocated Discrete Least Squares (CDLS) meshless [4], and similar. Although each of the listed methods possesses some unique properties, there is not much difference in their basic conception. In this book, the MLSM is employed as a generic name incorporating the basic principles of the meshless local strong form methods. The MLSM can also be understood as a meshless generalization of the FDM.

Another well-known weak form meshless method considered in this book is the Meshless Local Petrov Galerkin method (MLPG) [5]. It has been derived from the Weighted Residual Method (WRM) and follows similar principles as the FEM. As such, the MLPG can also be understood as a meshless generalization of the FEM.

It is claimed that meshless methods also perform well in situations with complicated geometry and a nonuniform node arrangement and that in comparison with mesh-based methods, they result in a smoother solution. However, since these methods are still under development, they have in most numerical simulations [6–8] only been demonstrated on simple geometries with regular node arrangements. Direct comparisons of different meshless methods are also rarely found in the literature. In this book, we want to contribute to a better understanding of meshless approaches, with emphasis on aspects of computer execution and implementation. The analyses focus first on the solution to a simple case formulated with diffusion equation, to assess the convergence rate, stability, and other basic properties of the methods. After initial evaluation, more complex cases are solved, i.e., fluid flow, semiconductor simulations, and solid mechanics problems. For assessment by simulation, we use uniform and nonuniform node arrangements and simple as well as more complicated geometries. The obtained results are also analyzed in terms of computer execution performance on modern computer architectures.

The book offers a broad insight into meshless methods from various points of view. The solution procedure is formulated in such a way that even readers with

only basic mathematical knowledge will probably be able to reproduce its results. Those readers who are more computer oriented will obtain a clear picture of meshless methodologies regarding the computer execution performance. Finally, researchers and engineers will find recipes for solving complex problems with a modern numerical methodology.

1.2 Why Solve PDEs?

Numerical analysis and computer modeling are becoming the basic tools of technological and scientific research. However, before we can discuss a numerical solution, an appropriate physical model is needed. Basically, each study of a natural or technological phenomenon begins with physical modeling that results in a more or less complex system of equations.

As illustration, let us present a simple example of the physical modeling of heat transfer. In the first step, the main characteristics of the phenomenon are described at the layman's level, i.e., the assumptions relating to the model are defined. The main assumptions for heat transfer are as follows:

- Heat is transported from hot regions to cold regions, i.e., heat flux ($\mathbf{j}$) is driven by the temperature gradient (∇T).
- The internal energy of a region can only change due to a difference between the inlet and the outlet heat fluxes (continuity).
- The temperature of a region is proportional to the internal energy stored in the region.

The next step is to rewrite the words in a more defined language for mathematical equations. The first assumption can be written as

$$\mathbf{j} = -\lambda \nabla T, \tag{1.1}$$

where λ stands for the material property describing the thermal conductivity, with, for example, $\lambda = 310\,\mathrm{W/mK}$ for gold, $\lambda = 2\,\mathrm{W/mK}$ for stone and $\lambda = 0.045\,\mathrm{W/mK}$ for wool. These numbers can be easily experienced in everyday life. For example, when you hold a piece of stone, you feel cold, since the warmth of your hands is quickly diffused away through a good thermal conductor. In contrast, wearing woolen gloves gives you a nice warm feel, since the heat generated by your body is preserved inside a good isolator. Equation (1.1) is also referred to as Fourier's law. The next assumption states the continuity of the internal energy. In other words, internal energy can neither be created nor destroyed, nor can it magically appear or disappear. It can only move in a continuous flow. This statement can be mathematically written as

$$\nabla \cdot \mathbf{j} + \frac{\partial E}{\partial t} = 0. \tag{1.2}$$

Finally, the last assumption gives us:

$$\frac{\partial E}{\partial t} = \rho c_p \frac{\partial T}{\partial t} \tag{1.3}$$

where ρ and c_p stand for the materials density and heat capacity, i.e., the ability of the material to store energy. Combining the three equations, we obtain

$$\frac{\partial T}{\partial t} = \frac{1}{\rho c_p} \nabla \cdot (\lambda \nabla T). \tag{1.4}$$

Equation (1.4) is well known as the diffusion equation and describes how the temperature distributes due its gradients [9]. The problem is fully determined by a set of boundary conditions (BCs) and an initial state. In general, there are three types of BCs:

- Dirichlet BC: the value of the field is known on the boundary, e.g., the temperature of the domain boundary is regulated by a thermostat, thereby keeping it at a constant value.
- Neumann BC: the value of the field derivative is known on the boundary, e.g., the domain is isolated and therefore the flux is zero, i.e., the normal derivative of the temperature is zero on the boundary.
- Robin BC: a linear combination of the field's value and the derivative is prescribed at the boundary, e.g., the energy flux through the boundary depends on the temperature at the boundary.

With Eq. (1.4), the heat transport can be described. For example, if we would like to assess the energy losses through the walls of a room, we would simply solve Eq. (1.4) with the appropriate material properties and boundary conditions.

1.3 The Background of the Numerical Solution

Equation (1.4), introduced in the previous section, is intuitive and simple and can be solved in a closed form on a rectangular domain. However, most of the PDEs that describe interesting natural phenomena cannot be solved in an exact way. In order to solve such problems, the continuous space is divided into discrete chunks, called nodes, i.e., a spatial and temporal discretization is made. Basically, the unsolvable differential equation is transformed into a system of algebraic equations that we can solve, resulting in an approximate solution to the problem at hand. There are many different methodologies for achieving this. The lowest level classification of the spatial discretization methods is into the strong and weak form methods. The difference between the strong and the weak form is in the core concept. The weak form methods minimize the residual between the unknown solution and the trial solution function, which is essentially the numerical solution of the problem on the

whole domain. On the other hand, the strong form seeks the solution only through an evaluation of the PDEs in the discretization nodes.

In the weak form, the trial function is constructed by shape functions that are defined by the position of the nodes and the type of the basis functions. The weak form methods also require a numerical integration of the shape functions and their derivatives to compute the local residuals. The evaluation of the trial solution is needed in the integration points, which are typically not collocated with the discretization nodes. The strong form methods, on the other hand, employ nodal trial functions. A nodal trial function is basically a local approximation of the considered field, valid only near the discretization nodes. It is a linear combination of the basis functions and can be easily manipulated. For example, to compute a derivative of the considered field, we have to construct the nodal trial function and derive it.

Finally, the goal of both approaches is to construct a global system of equations, i.e., for each spatial discretization node, one equation based on the PDE that we want to solve. If the solution is evolving over time, explicit or implicit approaches can be used for stepping forward with time. In the case of explicit stepping, usually reasonable only for strong form methods, the stepping towards a solution does not require any solution of the system, just a matrix vector multiplication. In contrast, implicit approaches usually enable faster stepping, but with the extra price of a global system solution in each time step.

The method is considered local when the manipulation of the field in a selected node requires information only from a small subset of the domain discretization nodes, often referred to as the support domain or the subdomain. The local approach results in a banded system that is easier and faster to solve and also offers a more convenient formulation for a parallel implementation on modern computers. The main drawback of the global approaches is the necessity for solving a global linear system represented by dense matrices. The conditioning of such a system is generally sensitive to the distribution of the domain nodes and to the parameters of the numerical method itself. The problem could become critical even with a relatively small number of nodes, e.g., 1000. The mitigation of the related problems has been attempted by domain decomposition [10], multigrid approaches and compactly supported basis functions [11], which represent a substantial complication of the method. It was demonstrated [12] that in the case of a local formulation being employed instead of the global one, the accuracy is not substantially degraded. The local formulation is much less sensitive and more effective with respect to the computational time. The global approach also requires a complex parallelized computer code and a significant amount of interprocessor communication. From the computational point of view, localization reduces the interprocessor communication, a common bottleneck with parallel algorithms [13].

Regardless of the complex formulation, computation on a coarse spatial discretization with fast convergence has been the ultimate advantage of the weak form methods in the past. However, the availability of modern parallel computing platforms is changing the situation. Computational simplicity is becoming a more important factor, even at the cost of larger numbers of iterations and discretization points [14], since a vast number of computing units can be used. From the point of view of

implementation and parallel execution, the simplest and the most effective numerical method is the explicit FDM, since it is completely local and simple to execute. However, the FDM is limited to a consideration of simple geometrical domains and restricted regarding its possibilities for upgrades. The FEM alleviates this drawback with various elements, but is not ideal for massively parallel computers. Its weak formulation requires a numerical integration and well-defined neighborhood relations between the discretization points [15]. As an alternative to the classic mesh-based methods, the meshless methods emerged in 1970s. For the FDM, the closest meshless alternative is the LRBFCM, while for the FEM, the meshless alternative is the MLPG. The conceptual difference between the mesh based and the meshless methods is in the definition of the nodal relations. The meshless methods fully define the nodal topologies solely through the internodal distances, while the mesh-based methods require additional information, i.e., the nodes need to be structured into polygons (mesh) that cover the whole computational domain.

1.4 Related Work

The classic mesh-based numerical methods, such as the FDM [16], the FVM [17], the FEM [18], and the Boundary Domain Integral Method (BDIM) [19], need discretization nodes that are organized into subsets of neighboring nodes for the determination of the polygons that cover the problem domain. The "mesh" denotes the connectivity between the corresponding neighboring nodes, obtained by some sort of spatial discretization. For unstructured, 3D geometries from the real world, the mesh construction is one of the most cumbersome and time-consuming steps in the entire numerical solution process [20], because it often requires significant assistance from the user. However, in the past few years, the coupling of Computer-Aided Design (CAD) and FEM analysis [21] has alleviated this problem. The approach is also referred to as isogeometric analysis and focused on the integration of the FEM into the conventional Non Uniform Rational Basis Splines (NURBS)-based CAD environments.

An alternative is the class of meshless methods (MLMs), where instead of using a mesh, a set of geometrically unconnected nodes can be used for the domain discretization. The MLMs originated in the 1970s, starting with the Smoothed Particles Hydrodynamics (SPH) [22]. Many variants of the meshless methods have been developed since then, based on different strong/weak formulations and approximation/interpolation techniques. Some examples are the Diffuse Element Method (DEM) [23], the Element Free Galerkin Method (EFG) [24], the Reproducing Kernel Particle Method (RKPM) [25], the Hp Cloud Method [26], the Partition of Unity FEM (PUFEM) [27], and the Meshless Galerkin Method using Radial Basis Functions (MGRBF) [8, 28, 29]. A simple integration mesh was needed even in the case of element free formulations. The weakly formulated meshless MLPG approach [30] has generated significant interest among researchers. Its theoretical background has been firmly developedand refined [31]. On the other hand, there are several variants

of the meshless methods based on strong form equations, e.g., the General FDM (GFDM) [32, 33], with an arbitrary mesh, the Radial Basis Function Collocation Methods (RBFCM) [34], LRBFCM [7], the Finite Point Method (FPM) [35], and DAM [36]. The intense development in the field of the meshless methods continues, which is also reflected in several relevant recent publications [36–43].

An important step in the meshless principle is the interpolation of the considered field. The interpolation [44, 45] or Moving Least Squares (MLS) approximation of nodal parameters from the support domain is used to create shape functions in the weak form methods or the nodal trial functions in the strong form methods. Even though the interpolation simplifies the implementation of the essential boundary conditions, there is no general method known to solve the possible singularities of the interpolation coefficient matrix in the case of the degenerated distribution of nodes or to ensure the continuity of the solution. On the other hand, the MLS is less influenced by an inappropriate nodal distribution in the support domain, since it is over determined by default. It is an essential approach that guarantees the locality of many meshless methods. The shape functions/nodal trial functions are built from the weighted contributions of a certain number of nearest nodes [46, 47]. The effects of the MLS support domain size have been studied in [48–50], with their impact on the efficiency, accuracy, and robustness of the meshless methodology.

The MLPG [5, 31] is the earliest representative of the truly meshless methods. Several variants of the MLPG differ in the construction of the trial functions and in the employed test functions. New methods that combine previous knowledge from the FEM and the BEM with the MLPG approach have been proposed in [51]. Several convergence issues have been addressed [28, 30, 52], mostly in connection with the structural analysis and often on smaller systems with a uniform distribution of nodes. The convergence of the MLPG methods depends on the MLS approximation accuracy and stability, and on the integration accuracy of the local weak form equations, all three depending on the distribution of the discretization nodes [53, 54]. Further studies have confirmed that the MPLG is a general concept that can be applied in various fields of applications from mechanics to Computational Fluid Dynamics (CFD) [55–59].

An alternative, and much simpler, local meshless approach, is the MLSM, e.g., the Diffuse Approximate Method (DAM) [60] or the LRBFCM [7]. Similar to the MLPG, in the MLSM, the nodal trial functions are constructed through the MLS or the collocation approach over the local support domain. In the MLSM, "local" refers to the locality of the nodal trial functions. The MLSM, as the name suggests, uses the strong form and, thus, no integration is required. All the differential operations are performed by the straightforward application of the differential operator on the nodal trial function.

Computational time is an important factor in numerical methods and is often not adequately addressed. The clock frequencies are approaching their physical limits and, therefore, the increased number of involved processing units is becoming attractive. Parallel computers, available today in most desktop computers or computer servers, can compensate for the lack of performance of a single computer, but only in

the cases where an efficient parallelization of the computational method is known. Various Application Programming Interfaces (APIs) for parallel programming are used to maximize the performance of parallel systems. Nowadays, the most widely used parallel programming libraries are based on a Message Passing Interface (MPI) in distributed memory systems, and APIs like POSIX threads programming (Ptreads) and Open MultiProcessing (OpenMP) in shared memory systems [61]. Moreover, the use of GPUs for solving parallel problems is continuing to spread. APIs that support parallel programming on GPUs are becoming more and more popular, like the Computer Unified Device Architecture (CUDA) and the Open Computing Language (OpenCL) [62, 63].

Chapter 2
Discretization and Formulation of Solution Approaches

Abstract Discretization principles as a foundation for numerical approaches and PDE solution methodologies are described, for space and time, from a bird's-eye view. Two PDE solution methodologies, based on the strong or the weak form, in mesh-based and meshless methods, are introduced briefly.

Keywords Strong form · Weak form · Discretization of time

2.1 Background

In the previous section, we showed how to formulate a simple physical model for heat transfer (see Eq. (1.4)). The formulated diffusion PDE depends on spatial and temporal variables. Now, we model a more general problem as:

$$\mathscr{L}u(\mathbf{x}) = g(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{2.1}$$

with boundary conditions:

$$\mathscr{B}u(\mathbf{x}) = h(\mathbf{x}), \qquad \mathbf{x} \in \Gamma, \tag{2.2}$$

where u is a continuous unknown solution, $\mathbf{x}$ is a vector of continuous independent variables, $\mathscr{L}$ and $\mathscr{B}$ are differential operators, g and h are known functions, Ω is the problem domain, and Γ is its boundary.

We continue by describing the functions in terms with which we will express the numerical solutions of the PDEs:

- The *basis functions* p_j are the members of the *basis*, a set of functions that spans the space of the employed interpolating or approximating functions. The typical basis functions are monomials, Gaussian, splines, etc.
- The *shape functions* ϕ_i are linear combination of the basis functions that can reconstruct arbitrary field u through an interpolation or approximation. The function can also be interpreted as a nodal approximation function of value 1 in node

R. Trobec and G. Kosec, *Parallel Scientific Computing*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-17073-2_2

$\mathbf{x}_i$ and 0 in all the other nodes from the global domain Ω. The shape functions are fully determined by the distribution of nodes and definition of basis functions. A smooth, hat-shaped weight function w_i can be applied to control the amount of nodals' impacts.

- The *nodal trial function* $\hat{u}_i$ of a node $\mathbf{x}_i$ is a parameterized function that approximates or interpolates the field u on the local support domain Ω_{Si}.
- The *trial function* $\hat{u}$ of the unknown solution u is a parameterized function that approximates or interpolates the field u on the global domain Ω.

To summarize, from suitable basis functions and nodal positions, we can create shape functions that are used to construct the trial functions. The next chapter describes in more details how this is done. Please note, that we will differentiate between $\hat{u}$ and u only where the difference must be stressed, otherwise u will be used for the approximate solution.

In most numerical methods for solving PDEs, the general strategy is to represent continuous unknown field u, e.g., temperature, pressure, stress, velocity, etc., with discrete values in a set of discretization points of independent variables, e.g., spatial coordinates, time, or others. The spatial discretization strategy relies on a distribution of N discretization points through the problem domain Ω for which a solution is sought. We will term the discretization points as nodes:

$$\mathbf{x}_i \in \Omega, \quad \text{for} \quad i \in \{1, \ldots, N\}, \tag{2.3}$$

in order to distinguish them from any other points in the domain, e.g., evaluation points for the calculation of numerical integrals, evaluation points for visualization, etc. The independent variables $\mathbf{x}$ and the approximate solution $\hat{u}$ are discretized using the set of nodes $\mathbf{x}_i$ and the corresponding nodal parameters $\hat{u}(\mathbf{x}_i) = u_i$. The solution in the nodes can encompass more variables, e.g., temperature, velocities, pressure, displacements, etc., in such cases the vector notation $\mathbf{u}(\mathbf{x_i})$ will be used.

The conceptual difference between mesh-based and meshless methods is in the way the discretization nodes are treated. In mesh-based methods, they are organized in a mesh before the solution procedure, using a priori knowledge about the neighboring nodes of $\mathbf{x}_i$ and the relations between them. In meshless methods, no a priori knowledge about the nodal topology is required. To determine the support domains, simple algorithms like Nearest Neighbors Search (NNS) can be used, either during the simulation (SPH) or in a preprocess phase (DAM, LRBFCM).

In mesh-based methods, the discretization nodes can be organized in a mesh of polygons, traditionally called *elements*. The mesh is usually determined by a list of elements with the corresponding ordered nodes. Note that for regular meshes with an isomorphic neighborhood of nodes (for example, the FDM), the list can be generated explicitly, i.e., the neighboring nodes are determined by an explicit function. Unfortunately, in many real cases with nonregular geometries in Ω, such