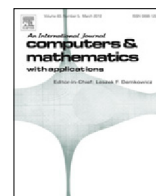




Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0



R. Vázquez

Istituto di Matematica Applicata e Tecnologie Informatiche del CNR, Via Ferrata, 1, 27100 Pavia, Italy

ARTICLE INFO

Article history:

Received 14 April 2016

Accepted 12 May 2016

Available online 11 June 2016

Communicated by D. Boffi

Keywords:

Isogeometric analysis

Finite element method

NURBS

B-splines

Matlab

Octave

ABSTRACT

GeoPDEs (<http://rafavzqz.github.io/geopdes>) is an Octave/Matlab package for the solution of partial differential equations with isogeometric analysis, first released in 2010. In this work we present in detail the new design of the package, based on the use of Octave and Matlab classes. Compared to previous versions the new design is much clearer, and it is also more efficient in terms of memory consumption and computational time.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

GeoPDEs [1] is a software suite for the solution of partial differential equations using isogeometric analysis (IGA), a Galerkin discretization method based on splines [2]. Written in Octave [3] and fully compatible with Matlab, the package was first released in 2010 under a GNU/GPL license, with the intention to serve both as a research tool, for fast prototyping of new ideas in IGA, and as a teaching tool, to introduce isogeometric methods to other researchers and students interested in the topic.

In the last five years new IGA software has been released, with at least one more package written in Matlab, focused on computational solid mechanics [4], and several libraries written in C++ or C looking for more efficiency. In particular, the `igatools` library [5] provides a general and dimension independent implementation of IGA; the `G+Smo` library [6] is, up to the author's knowledge, the only existing open-source software for IGA which includes hierarchical splines; and `PetIGA` [7] is a C implementation of isogeometric methods based on PETSc, which has been recently extended to include the curl- and div-conforming spline discretizations [8]. A more detailed list of available software for IGA can be found in [4]. Despite the appearance of these new libraries, GeoPDEs remains one of the most successful available packages for IGA.¹ The easiness to learn from an Octave/Matlab² code, the detailed documentation (accessible with the `help` command) and the long list of existing examples have surely contributed to this success.

Since GeoPDEs is a fundamental tool for the author's own research, the package is under continuous development, and it has evolved during the last years to a more efficient and clearer design, sometimes inspired by the developments in other IGA libraries, and sometimes by the needs and the contributions of GeoPDEs users. These changes have helped to maintain

E-mail address: vazquez@imati.cnr.it.

¹ Around 3000 downloads in the last three years.

² In the sequel we will mention Octave to refer indistinctly to both Octave and Matlab, unless we explicitly indicate it.

the package as a tool suitable for innovative research. As a drawback, they have made the original article [1] obsolete and unsuitable to understand its current design, making necessary a new reference containing an explanation of the new design.

Another important motivation for the preparation of this paper is related to the future presentation of new features and methods that are being implemented within GeoPDEs. In particular, the author is currently working, in collaboration with E.M. Garau, on an extension of GeoPDEs to include adaptivity based on hierarchical splines [9]. As far as we know, this will be the first open-source software for IGA adaptivity in Octave, and we expect (or strongly hope) that it will have a great impact for the future development of IGA. Obviously, this extension is based on the current design of the code, which makes necessary to provide a valid reference to potential users to understand the design of the package.

The new version of GeoPDEs is not a simple modification of the first release of the package, instead, it is a complete redesign of the code. Rather than simply explaining the differences with the first version, in this work we present the complete design of GeoPDEs 3.0 in a clean form, with the aim of making this paper a comprehensive reference for the current release of GeoPDEs and to give the reader a global overview of the package. The main features of GeoPDEs are the following: (i) dimension independent implementation, inspired by [5], in such a way that the same code is valid for curves, surfaces and volumes; (ii) implementation of vector fields with div-conforming and curl-conforming discretizations, also in multipatch domains; (iii) use of Octave classes instead of structures, avoiding the precomputation of fields that caused memory problems in previous versions; (iv) new multipatch classes, that automatically manage the strong C^0 gluing of basis functions and the matrix assembly in multipatch geometries in a user-friendly way; (v) inclusion of some properties into the classes, allowing to reduce the number of classes with respect to the original structures in previous versions.

The paper is organized as follows: in Section 2 we briefly introduce IGA in an abstract form, mainly to fix the notation, and we apply it to the solution of the Poisson problem. Based on this notation we explain in detail the design of the code in Section 3. We then present in Section 4 the implementation of discrete spaces for the approximation of vector fields, and in Section 5 we detail the new classes for the construction of isogeometric discrete spaces in geometries formed by multiple patches. Finally, in Section 6 we show a comparison of the performance of the current version of GeoPDEs with the one presented in [1]. The paper is complemented with several appendices with practical information for users of the software, and a webpage (<http://rafavzqz.github.io/geopdes>) to download the software.

2. The basics of isogeometric analysis

In this section we recall the main concepts of IGA, mainly to fix the notation, assuming that the reader has a basic knowledge of the method. The interested reader can find more details about the topic in [2,10]. We start this section with an abstract framework for IGA, already introduced in [1], and that will serve us later to explain the design of the package. We then present the definitions of B-splines and NURBS, and show how this abstract IGA framework is applied to a simple model problem, the Poisson problem, with a NURBS discretization. For the sake of clarity, in this section we restrict ourselves to the single patch case, that is, the domain is defined as the image of the n -dimensional unit domain. A more complex framework for multipatch domains is explained in Section 5.

2.1. Isogeometric analysis: an abstract framework in a single patch domain

Similar to the finite element method, the goal of IGA is the numerical approximation of the solution of partial differential equations (PDEs) with a Galerkin method.³ The boundary value problem is written in variational form, and we seek a discrete solution by solving the variational problem in a finite-dimensional space with good approximation properties.

We assume that the domain of our problem is given by a parametrization, that is, we have a *parametric domain* $\widehat{\Omega} = (0, 1)^n$, and a *physical domain* $\Omega = \mathbf{F}(\widehat{\Omega}) \subset \mathbb{R}^r$, with $n \leq r$, which is defined through the *parametrization* \mathbf{F} . Notice that this parametrization is not required to be defined as a NURBS. The Jacobian of \mathbf{F} is given by an $r \times n$ matrix, that we denote by $J_{\mathbf{F}}$.

Let V be a Hilbert space, to which the solution of the continuous problem belongs, and $V_h \subset V$ a discrete space where we look for an approximate solution. Given a bilinear form $a : V \times V \rightarrow \mathbb{R}$ and a function $f \in L^2(\Omega)$, the variational formulation of our discrete problem reads: Find $u_h \in V_h$ such that

$$a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h, \quad (1)$$

where (\cdot, \cdot) represents the L^2 -inner product. In a general IGA context the *approximation space* V_h is defined as

$$V_h := \{v_h = \iota_{\mathbf{F}}(\widehat{v}_h), \widehat{v}_h \in \widehat{V}_h\} \quad (2)$$

where \widehat{V}_h is a discrete space in the parametric domain $\widehat{\Omega}$, usually a space of splines or NURBS, and $\iota_{\mathbf{F}}$ is a suitable *push-forward* defined from the parametrization \mathbf{F} .

Assuming that the parametrization is regular enough, given a *basis* $\widehat{\mathcal{B}} = \{\widehat{v}_j, j = 1, \dots, N_h\}$ of the finite-dimensional space \widehat{V}_h , with $N_h = \dim \widehat{V}_h$, we can define a basis of V_h applying the push-forward to these basis functions, that is,

³ The design of GeoPDEs is based on elements, and is not well suited for IGA collocation methods.

Download English Version:

<https://daneshyari.com/en/article/470253>

Download Persian Version:

<https://daneshyari.com/article/470253>

[Daneshyari.com](https://daneshyari.com)