Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 285 (2015) 817–828
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization

P. Antolin[a], A. Buffa[b], F. Calabrò[c], M. Martinelli[b], G. Sangalli[d,b,*]

[a] *Dipartimento di Ingegneria Civile ed Architettura, Università degli Studi di Pavia, Italy*
[b] *Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes" del CNR, Pavia, Italy*
[c] *DIEI, Università degli Studi di Cassino e del Lazio Meridionale, Italy*
[d] *Dipartimento di Matematica, Università degli Studi di Pavia, Italy*

Received 1 July 2014; received in revised form 2 December 2014; accepted 4 December 2014
Available online 15 December 2014

Highlights

- We discuss the use of the sum-factorization for the calculation of the integrals arising in Galerkin isogeometric analysis.
- We give an estimate of the quadrature computational cost and compare with the standard approach.
- We perform numerical tests.
- Sum-factorization significantly reduces the quadrature computational cost.

Abstract

In this paper we discuss the use of the sum-factorization for the calculation of the integrals arising in Galerkin isogeometric analysis. While introducing very little change in an isogeometric code based on element-by-element quadrature and assembling, the sum-factorization approach, taking advantage of the tensor-product structure of splines or NURBS shape functions, significantly reduces the quadrature computational cost.

Keywords: Numerical integration; Isogeometric analysis; Splines; NURBS; Sum-factorization

1. Introduction

Isogeometric analysis (IGA) is a computational technique for the solution of boundary value problems. It is recent and at the same time well known in the computational engineering academic community, as an extension of the classical Finite Element Method (FEM). IGA was proposed in the seminal paper [1], based on the idea of using the functions adopted in Computer Aided Design (CAD), that is, splines and Non-Uniform Rational B-Splines (NURBS),

* Corresponding author at: Dipartimento di Matematica, Università degli Studi di Pavia, Italy.
E-mail addresses: pablo.antolinsanchez@unipv.it (P. Antolin), annalisa@imati.cnr.it (A. Buffa), calabro@unicas.it (F. Calabrò), martinelli@imati.cnr.it (M. Martinelli), giancarlo.sangalli@unipv.it (G. Sangalli).

http://dx.doi.org/10.1016/j.cma.2014.12.013
0045-7825/

not only to describe the domain geometry, but also to represent the numerical solution of the problem, in the isoparametric framework. For the interested reader, we refer to the book on IGA [2]. A recent overview on the mathematical aspects of IGA is [3], that covers the known mathematical theory of IGA but also contains an updated bibliography with references to the major contributions and applications of IGA in various engineering fields.

One of the interesting features of IGA, compared to high order FEM, is that it allows for higher global regularity of the shape functions, up to C^{p-1} inter-element continuity for p-degree splines and NURBS. This leads to a higher accuracy per degree-of-freedom (see [4,5]), improved spectrum properties of the discrete operators (see [6]), and the possibility of constructing smooth discretizations of the fundamental structures of the differential operators (such as De Rham diagrams, see e.g. [7,8]).

IGA can be implemented re-using the existing finite element technology. This may be not the most efficient way to use IGA but surely is one key reason of its fast diffusion and the easiest way to apply IGA on complex problems. In particular, the construction of the matrix of the linear system arising in a Galerkin isogeometric method is typically made by the element-by-element quadrature and assembling as in FEMs. However for high regular and high degree ($p \geq 3$) splines and NURBS, it is experienced that most of the CPU time goes in the quadrature and assembling itself. This may be understood comparing IGA with C^0 to C^{p-1} p-degree splines (or, equivalently, FEM and typical IGA) on the same mesh: element-wise quadrature has the same computational cost in the two cases, even though the C^{p-1} case results in a much smaller linear system. The high cost of quadrature has motivated the research on quadrature rules that keep into account the interelement regularity of IGA functions, see [9–11], improving efficiency w.r.t. Gauss quadrature. In this paper we consider another significant improvement: one can exploit the tensor-product structure of multivariate splines by adopting the so called *sum-factorization*, a well-known technique for spectral elements or some high-degree finite elements (see e.g. [12–14]), but never used with IGA, at our knowledge.

The aim of this paper is to discuss and benchmark the use of sum-factorization in IGA. We show that there is a clear advantage versus the standard quadrature approach, and show that the cost of quadrature (by sum-factorization) is balanced with the cost of the linear solver for high degree IGA. We also discuss the implementation of the proposed sum-factorization within our isogeometric library `igatools` [15]. We do not consider parallel implementation though this is clearly a key ingredient for a modern and efficient isogeometric code (see for example [16]).

There are other possibilities to circumvent the element-by-element quadrature issue that however require a change of paradigm. For example, if the mesh is uniform, one can efficiently and directly compute the entries of the linear system matrix (see [17–20]) or switch from Galerkin to a collocation formulation [21,22].

The outline of the paper is as follows. In Section 2 we set up the notation and briefly describe the setting of an academic problem. Section 3 introduces the sum-factorization algorithm and discuss its computational cost in terms of the degree p. Section 4 is devoted to the numerical testing and comparison with other strategies. Finally, we draw conclusions in Section 5. An Appendix is included in order to describe the treatment of the linear elasticity stiffness matrix.

2. Preliminaries

We consider the elliptic problem

$$\begin{cases} -\mu\nabla^2 u + \sigma u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{2.1}$$

as a model problem. Its Galerkin approximation on a discrete space V requires the computation of the following matrix entries:

- the mass matrix (or mass integrals)

$$M_{i,j} = \int_\Omega \sigma(\boldsymbol{x}) R_{\boldsymbol{i}}(\boldsymbol{x}) R_{\boldsymbol{j}}(\boldsymbol{x})\, d\boldsymbol{x}; \tag{2.2}$$

- the stiffness matrix (or stiffness integrals)

$$S_{i,j} = \int_\Omega \mu(\boldsymbol{x}) \nabla R_{\boldsymbol{i}}(\boldsymbol{x}) \cdot \nabla R_{\boldsymbol{j}}(\boldsymbol{x})\, d\boldsymbol{x}; \tag{2.3}$$

where $R_{\boldsymbol{i}}$ and $R_{\boldsymbol{j}}$ denote two basis function in V, $\mu : \Omega \to \mathbb{R}$ and $\sigma : \Omega \to \mathbb{R}$.