Applied Mathematics and Computation 301 (2017) 201–213

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Extending linear finite elements to quadratic precision on arbitrary meshes

Daniel Duque[a,*], Pep Español[b], Jaime Arturo de la Torre[b]

[a] *Canal de Experiencias Hidrodinámicas (CEHINAV), ETS Ingenieros Navales, Avda. Arco de la Victoria 4, Madrid E-28040, Spain*
[b] *Departamento de Física Fundamental, UNED, Apartado 60141, Madrid 28080, Spain*

ARTICLE INFO

Keywords:
Finite element method
Quadratic precision
Linear precision
Laplacian

ABSTRACT

The method of finite elements can provide discrete versions of differential operators in general arrangements of nodes. Linear finite elements defined on a triangulation, like Delaunay's, constitute a typical route towards this aim, as well as being a tool for interpolation. We discuss a procedure to build interpolating functions with quadratic precision from this functional set. The idea is to incorporate (extend) the products of the original functions in order to go from linear precision (the property that linear functions are reconstructed exactly) to quadratic precision (quadratic functions are reconstructed exactly). This procedure is applied to the standard Galerkin approach, in order to study the Poisson problem and the direct evaluation of the Laplacian. We discuss convergence of these problems in 1D and 2D. The extended method is shown to be superior in 2D, featuring convergence for the direct evaluation of the Laplacian in distorted lattices.

1. Introduction

In many areas of science and engineering the spacial domain is discretized. The numerical solution of differential equations, for example, requires versions of the differential operators which are constructed after discretizing space. The failure or success of a numerical scheme depends on the accuracy of this procedure.

Here, we are primarily interested in irregular (unstructured) meshes. For example, in Lagrangian computational fluid dynamics (CFD) methods the mesh is carried with the flow and distorted. These methods are specially suited for the study of violent flows involving free surfaces because of their natural adaptivity to the domain where the fluid actually is. Examples of Lagrangian methods include the Smoothed Particle Hydrodynamics [1], the closely related Moving Particle Semi-Implicit [2], and the Dissipative Particle Dynamics model for mesoscopic fluctuating flows [3]. Other methods that employ moving Delaunay triangulations or their dual (the Voronoi tessellation) to generalize either the Finite Volume Method [4–6] or Finite Element Method [7–12] have also been considered to address flow from a Lagrangian perspective. This is the field in which the ideas described here were derived, but our framework is quite general. For example, it could alleviate the mesh design of Eulerian methods. Here, we address the simple task of evaluating the Laplacian of a function, and the inverse problem (the Poisson problem).

The primary motivation for this work can be found in Fig. 8, that shows results for the evaluation of the Laplacian of a field. Standard linear finite elements (circles) produce a discrete operator that is nicely converging (left figure) as the

* Corresponding author.
E-mail address: daniel.duque@upm.es (D. Duque).

http://dx.doi.org/10.1016/j.amc.2016.12.010
0096-3003/

number of points is increased, on a regular grid. However, they dramatically fail to converge on a random grid, even for a small distortion (10%) from the regular grid (right figure). This fact has encouraged us to find ways in which this situation could be improved. As we discuss, an extension of finite element linear basis functions that is able to reconstruct quadratic functions (results are in squares symbols in Fig. 8) does converge for distorted lattices. Our approach is different from high order finite element methods, since we are not able to place our nodes freely, as the mesh in Lagrangian methods is prescribed to move according to the flow field. Rather, we introduce a new functional set using the same nodes (therefore not increasing the degrees of freedom), but with an increased support.

The paper is organized as follow. In Section 2 we describe the general framework, which we will refer to as the “extended method” and explain in detail its implementation in one and two dimensions. In Section 2.5 we use the extended basis function in order to obtain discretized versions of the differential operators. We follow a standard procedure, along the familiar lines of the method of weighted residuals and the Galerkin method [13,14]. The method is tested in Section 3, both in 1D and 2D. Some conclusions are presented in Section 4. The relevant overlap integrals that have been used are given in Appendix A.

2. Theory

2.1. Original set

For an arbitrary arrangement of N nodes labeled by the index μ, let us consider a set of functions $\{\phi_\mu\}$ that reconstruct constant functions exactly (a property known as partition of unity, if the functions are positive):

$$\sum_{\mu=1}^{N} \phi_\mu(\mathbf{r}) = 1. \tag{1}$$

We will suppose that linear functions are also exactly reconstructed by this set (a property known as the local co-ordinate):

$$\sum_{\mu=1}^{N} \phi_\mu(\mathbf{r})\mathbf{r}_\mu = \mathbf{r}. \tag{2}$$

These two properties combined define “linear precision”. They are satisfied by many functional sets, including the well-known linear finite elements (FE) shape functions, natural neighbor interpolants (either Sibsonian or otherwise) [15], and LME and SME interpolants [16,17].

In the present paper, we will use linear FE shape functions. In 1D, the nodal functions for node μ is continuous and piece-wise linear, with $\phi_\mu(x)$ going from 0 at $x_{\mu-1}$ to 1 at x_μ, back to 0 at $x_{\mu+1}$. They are zero outside the segment $(x_{\mu-1}, x_{\mu+1})$. Similarly, in 2D $\phi_\mu(\mathbf{r})$ is a pyramid of height 1 at node μ with straight edges that connect the apex with the neighboring nodes. These neighbors are determined by an underlying 2D triangulation; there are many possible ones, but we choose, as is customary, the unique (except for possible degeneracies) Delaunay triangulation [18], thus defining natural neighbors.

The FE basis functions do *not* comply with quadratic precision in general:

$$\sum_{\mu=1}^{N} \mathbf{r}_\mu\mathbf{r}_\mu\phi_\mu(\mathbf{r}) \neq \mathbf{r}\mathbf{r}. \tag{3}$$

Here, the notation **ab** means a tensor formed by the dyadic product of vectors **a** and **b**. In terms of Cartesian components, $(\mathbf{ab})_{ij} := \mathbf{a}_i\mathbf{b}_j$.

2.2. Extension to quadratic precision

We here consider an “extended” set of basis functions derived from the FE $\{\phi_\mu\}$ basis set, with functions:

$$\psi_\mu(\mathbf{r}) := \phi_\mu(\mathbf{r}) + \sum_{\nu\sigma} A_{\mu\nu\sigma}\phi_\nu(\mathbf{r})\phi_\sigma(\mathbf{r}), \tag{4}$$

i.e. a linear combination of the original functions and their pair-wise products. The inclusion of these products extends the spatial support of the original functions (hence their name; the number of functions in each set is the same.)

The coefficients will be taken to be symmetric with respect to the last two indices: $A_{\mu\nu\sigma} = A_{\mu\sigma\nu}$, and are determined so as to fulfill not only the equivalent of Eqs. (1) and (2), but also to comply with quadratic precision:

$$\sum_{\mu=1}^{N} \mathbf{r}_\mu\mathbf{r}_\mu\psi_\mu(\mathbf{r}) = \mathbf{r}\mathbf{r}. \tag{5}$$