A new strategy for finite element method adaptation

Piotr Opioła
Institute of Computer Science
Faculty of Mathematics and Computer Science
Jagiellonian University
ul. prof. Stanisława Łojasiewicza 6, 30-348 Kraków, Poland

Abstract

There are several adaptive approaches to finite element method, including the r-method, the p-method and the h-method. The adaptive methods are used to optimize the mesh and the approximation function by minimizing a monitor function. The monitor function usually consists of an approximation error estimate and some chosen measures of the mesh quality, such as smoothness or orthogonality. In the paper, we present a new approach, in which the monitor function consists of an approximation error estimate, the number of mesh points and a numerical error estimate. The optimization is performed by adding and removing the mesh points, and by changing the approximation order. The method has been tested on several boundary and initial problems, including the Laplace's equation, the heat equation and the wave equation.

Keywords: adaptive finite element method, partial differential equations

1. Introduction

The finite element method is a powerful tool for solving partial differential equations, mainly appreciated for being general and flexible. However, the process of adjusting the method to a particular problem is usually difficult and exacting. To make this process more convenient, an automatic mechanism is necessary. Such a mechanism is provided by adaptive finite element methods, which have been developed for years [1, 2], still being improved.

The novelty of our method is in the intuitive formulation of the monitor function, flexibility of the approximation function, and effective mesh generation algorithm for higher dimensions. Similarly to e.g. [3], the monitor function consists of an approximation error estimate and measures of the mesh quality. The difference is that orthogonality and smoothness as the mesh quality measures are replaced by the number of mesh points and a numerical error estimate. Such a formulation helps in controlling the method efficiency with two meaningful parameters: the expected approximation error and the expected number of points.

2. Problem description

We consider three first-order BVPs

$$\nabla u(x) = f(x), \text{ for } x \in \Omega \subset \mathbb{R}^n, \tag{1}$$

$$\mathrm{div} u(x) = f(x), \text{ for } x \in \Omega \subset \mathbb{R}^n, \tag{2}$$

$$\nabla_v u(x) = f(x), \text{ for } x \in \Omega \subset \mathbb{R}^n, \tag{3}$$

where ∇_v is the directional derivative in the direction v. We seek a solution $u : \Omega \to \mathbb{R}$ in the $\mathcal{C}^1$ class. The boundary conditions are given by

$$u(x) = g(x), \text{ for } x \in \Gamma = \partial\Omega, \tag{4}$$

We define a mesh as a finite set $\mathcal{M} = \{x_i\}_{i=1\dots N} \subset \Omega$ and a neighborhood function $\mathcal{N} : \Omega \to \mathcal{P}(\mathcal{M})$. The neighborhood function maps a point $x \in \Omega$ into a nonempty set of its neighbors. The domain is divided into elements. Two points x, y belong to the same element iff $\mathcal{N}(x) = \mathcal{N}(y)$. Having a mesh $(\mathcal{M}, \mathcal{N})$, we define the approximation $\tilde{u} : \Omega \to \mathbb{R}$ as

$$w_i(x) = 1 - (|\mathcal{N}(x)| - 1)\frac{\|x - x_i\|}{\sum_{x_j \in \mathcal{N}(x)} \|x - x_j\|}, \tag{5}$$

$$\tilde{u}(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} w_i(x)u_i + w_i(x)^{p_i} r_i(u_i + q_i \cdot (x - x_i))}{\sum_{x_i \in \mathcal{N}(x)} w_i(x) + w_i(x)^{p_i} r_i}, \tag{6}$$

where $p_i \in \mathbb{R}$, $p_i > 0$, $r_i \in \mathbb{R}$, $r_i \geq 0$, $u_i \in \mathbb{R}$, $q_i \in \mathbb{R}^n$ are the approximation parameters. The parameters, as well as the mesh points $\mathcal{M}$, are sought to minimize the monitor function E, defined as

$$E = \frac{1}{\alpha}E_\varepsilon + \frac{1}{\beta}E_N + E_\delta, \tag{7}$$

where

$$E_\varepsilon = \max_{x \in \Omega} |\tilde{u}(x) - u(x)|, \tag{8}$$

$$E_N = N, \tag{9}$$

$$E_\delta = \max_{i \neq j} \frac{1}{\|x_i - x_j\|}. \tag{10}$$

The parameters α and β have the meanings of the expected error and the expected number of mesh points respectively. The parameters provide a convenient way to control the balance between precision and efficiency of the method.

3. Algorithm

The method takes the following steps

1. Generate a regular mesh in Ω
2. Refine the mesh by adding a new point in the center of every element
3. Solve Eqn. (1), (2) or (3) numerically on the current mesh
4. Coarsen the mesh by removing unimportant points

5. If the monitor function changed significantly, go to step 2

Step 3 of the algorithm depends on the equation to be solved. For Eqn. (1) it is done in the following steps

1. $u_i \leftarrow g(x_i)$, $x_i \in \Gamma$
2. $q_i \leftarrow f(x_i)$, $x_i \in \Omega$
3. $u_i \leftarrow \tilde{u}^\star(x_i)$, $x_i \in \Omega$
4. If any of the parameters changed significantly, go to step 3

The star * denotes values calculated for $\mathcal{M}\backslash\{x_i\}$, that is without considering x_i as a mesh point. In the case of the second problem, given by Eqn. (2), the above algorithm is changed into

1. $u_i \leftarrow g(x_i)$, $x_i \in \Gamma$
2. $q_i \leftarrow \nabla\tilde{u}(x_i)$, $x_i \in \Gamma$
3. $q_i \leftarrow q_i + f(x_i) - (q_i \cdot \vec{1})\vec{1}$, $x_i \in \Omega$
4. $u_i \leftarrow \tilde{u}^\star(x_i)$, $x_i \in \Omega$
5. If any of the parameters changed significantly, go to step 2

The notation $\vec{1}$ refers to the vector of ones. In the case of the third problem, given by Eqn. (3), the algorithm takes the form

1. $u_i \leftarrow g(x_i)$, $x_i \in \Gamma$
2. $q_i \leftarrow \frac{f(x_i)}{v \cdot v} v$, $x_i \in \Omega$
3. $u_i \leftarrow \tilde{u}^\star(x_i)$, $x_i \in \Omega$
4. If any of the parameters changed significantly, go to step 3

Coarsening the mesh involves adjusting the parameters p_i and r_i. This is done in the following steps

1. Find such p_i, r_i that minimize $|\tilde{u}(x_i) - \tilde{u}^\star(x_i)|$, $x_i \in \Omega$
2. Find x_i for which $\Delta E = E^\star - E$ is the lowest
3. If $\Delta E < 0$ then remove x_i and go to step 2

4. Results

The first test case is the Laplace's equation split into two first-order equations

$$\mathrm{div}\phi = 0 \tag{11}$$

$$\nabla u = \phi \tag{12}$$

The domain is a cube $\Omega = [0,1]^n$, $n = 2$. The boundary conditions for Eqn. (12) are given by $u(x) = x_1^2$, $x \in \Gamma$, where x_1 is the first coordinate of x. Equation (11) is solved with the boundary conditions $\phi(x) = \nabla\tilde{u}(x)$, $x \in \Gamma$, where $\tilde{u}$ is the current solution of Eqn. (12). Equation (12) is solved with ϕ being the current solution of Eqn. (11). The equations are solved repeatedly until the changes in $\tilde{u}$ are negligible.

Table 1: Numerical results for the Laplace's equation

α, β	E_ε	Real error	Iterations	N
1E-10, 1	1E-6	0.3086	657	241
0.01, 1	0.0773	0.6026	72	17
1E-10, 1000	0.0001	2.398	425	330
0.01, 1000	0.0025	9.573	219	73

The second test case is the heat equation split into three equations

$$\frac{\partial}{\partial t} u = \mu\,\mathrm{div}\phi \tag{13}$$

$$\mathrm{div}\phi = \frac{1}{\mu}\frac{\partial}{\partial t} u \tag{14}$$

$$\nabla u = \phi \tag{15}$$

The domain is consisted of a spatial domain and a time domain $\Omega = \Omega_x \times [0,1]$. The spatial domain is a cube $\Omega_x = [0,1]^{n-1}$, $n = 3$. The spatial boundary conditions for Eqn. (13) and (15) are as previously for Eqn. (12). The spatial boundary conditions for Eqn. (14) are the same sa previously for Eqn. (11). The time boundary conditions are $u(x,t) = 0$, $\phi(x,t) = 0$ for $x \in \Omega\backslash\Gamma$, $t = 0$. As previously, LHS of each equation contains the variable, and RHS of each equation is assumed given. The equations are solved repeatedly until the changes in $\tilde{u}$ are negligible.

Table 2: Numerical results for the heat equation

α, β	E_ε	Real error	Iterations	N
1E-10, 1	1E-7	0.2112	5000	2018
0.01, 1	0.0639	0.597	661	66
1E-10, 1000	1E-7	0.0769	6493	2922
0.01, 1000	0.0003	1.0587	1584	585

The third test case is the wave equation split into four equations

$$\frac{\partial}{\partial t}\rho = \mu\,\mathrm{div}\phi \tag{16}$$

$$\mathrm{div}\phi = \frac{1}{\mu}\frac{\partial}{\partial t}\rho \tag{17}$$

$$\nabla u = \phi \tag{18}$$

$$\frac{\partial}{\partial t} u = \rho \tag{19}$$

The domain and the initial conditions are the same as previously. The boundary conditions for Eqn. (17) and (18) are the same as previously for Eqn. (14) and (15) respectively. The spatial boundary conditions for Eqn. (16) are $\rho(x,t) = \frac{\partial}{\partial t}\tilde{u}(x)$, $x \in \Gamma$, $t \in [0,1]$. The boundary conditions for Eqn. (19) are the same as for (18). The equations are solved repeatedly until the changes in $\tilde{u}$ are negligible.

Table 3: Numerical results for the wave equation

α, β	E_ε	Real error	Iterations	N
1E-10, 1	1E-5	0.2579	5050	2836
0.01, 1	0.0238	0.5496	664	63
1E-10, 1000	1E-6	0.1162	6289	3508
0.01, 1000	0.0002	2.4803	1658	612

The numerical results, including the error estimate, the real error, the number of iterations and the number of mesh points, are presented in Tables 1, 2 and 3. The method has been tested for various values of the expected approximation error α and the expected number of mesh points β.

References

[1] E. Kita, “Error estimation and adaptive mesh refinement in boundary element method, an overview,” *Engineering Analysis with Boundary Elements*, vol. 25, no. 7, 2001.

[2] K. Segeth, “A review of some a posteriori error estimates for adaptive finite element methods,” *Mathematics and Computers in Simulation*, vol. 80, no. 8, pp. 1589–1600, 2010.

[3] G. Bono and A. Awruch, “An adaptive mesh strategy for high compressible flows based on nodal re-allocation,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 30, no. 3, pp. 189–196, 2008.