Applied Numerical Mathematics 60 (2010) 816–832

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A new stabilized finite element method for shape optimization in the steady Navier–Stokes flow ☆

Zhiming Gao [a,*], Yichen Ma [b]

[a] Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009-26, Beijing, 100088, PR China
[b] School of Science, Xi'an Jiaotong University, Shaanxi, 710049, PR China

ARTICLE INFO

Article history:
Received 3 April 2009
Received in revised form 12 April 2010
Accepted 17 April 2010
Available online 6 May 2010

Keywords:
Discrete adjoint method
Shape optimization
Stabilized finite element method
Gradient algorithm
Navier–Stokes equations

ABSTRACT

This paper investigates shape optimization of a solid body located in Navier–Stokes flow in two dimensions. The minimization problem of total dissipated energy is established in the fluid domain. The discretization of Navier–Stokes equations is accomplished using a new stabilized finite element method which does not need a stabilization parameter or calculation of high order derivatives. We derive the structures of discrete Eulerian derivative of the cost functional by a discrete adjoint method with a function space parametrization technique. A gradient type optimization algorithm with a mesh adaptation technique and a mesh moving strategy is effectively formulated and implemented.

1. Introduction

The optimal shape design of a body subjected to the minimum viscous dissipated energy has been a challenging task for a long time, and it has been investigated by several authors. For the computation of such problems, the gradient type optimization algorithms with adjoint methods are mostly utilized. Different methods exist to obtain the Eulerian derivative of the cost functional. For instance, Pironneau in [21,23] computes the derivative of the cost functional using normal variation approach; Murat and Simon [22] use the formal calculus to deduce an expression for the derivative; Bello et al. in [2,3] considered this problem theoretically in the case of Navier–Stokes flow by the formal calculus.

Based on the order of discretization and derivation of the adjoint equations, the adjoint method can be classified as continuous adjoint method and discrete adjoint method, or optimize-then-discretize method and discretize-then-optimize method. In the continuous adjoint method, one obtains the adjoint equations at the continuous level and then discretize the result; in the discrete adjoint method, one first discretizes the continuous state equations to obtain a set of discrete state equations and then differentiates the discrete approximate state equations to obtain the discrete adjoint equations. It is pretty common for stabilized methods that the discretization of the optimality conditions and the optimization of the discrete problem are not equivalent (see [4,6]). The shape gradients obtained by the two methods are not the same, but as the grid sizes go to zero, discrete adjoint variables and shape gradients of the discretized cost functionals all converge to the same solution. The discrete adjoint method constructs an exact gradient of the discretized cost functional and therefore there do not arise the inconsistencies between the computed gradient and the discretized functional. For the detailed discussion about the two methods, see Gunzburger [15]. Recently, Yagi and Kawahara in [29,30] study the optimal shape design for Navier–Stokes flow with boundary conditions containing the pressure using a discrete adjoint approach. However, their

☆ This work was funded by Laboratory of Computational Physics.
* Corresponding author. Fax: +86 10 62010108.
E-mail addresses: gao@iapcm.ac.cn (Z. Gao), ycma@mail.xjtu.edu.cn (Y. Ma).

0168-9274/$30.00 © 2010 Published by Elsevier B.V. on behalf of IMACS.
doi:10.1016/j.apnum.2010.04.008

proposed algorithm converges slowly. Katamine et al. [18] use the continuous adjoint approach with the formula of material derivative to study such problem involving velocity–pressure type boundary conditions with Reynolds number up to 100. Gao et al. [12–14] employ the continuous adjoint method to study shape sensitivity analysis for Stokes and Navier–Stokes flows.

In this paper, a gradient type optimization algorithm for steady Navier–Stokes flows with a discrete adjoint method and function space parametrization technique is proposed and implemented in two dimensions. A new stabilized finite element formulation is utilized for solving Navier–Stokes equations. It is based on two local Gauss integrations by using the lowest equal order pair of finite elements. The new method has some advantages: stabilization parameter-free, avoiding the calculation of high order derivatives (see He and Li [16,19] and references therein). For the study of the discrete shape gradient of the discretized cost functional, we will use the function space parametrization technique introduced by Delfour and Zolésio [9]. An unstructured triangular grid is utilized for complex geometries, a mesh adaptation technique and mesh movement strategy are employed during the optimization cycle. Finally, we present some numerical results and compare them for different values of the Reynolds number and various stabilized element pairs. The stabilized method is about 20% more efficient than MINI and Taylor–Hood elements.

In conclusion, the main advantages of our method are as follows:

- The stabilization term in our method can avoid the requirement of the inf–sup condition, and we only need to use low order elements.
- During the optimization, the formulae including the stabilization term are easy to code.
- Our optimization has less computational cost than optimizations using other stabilized methods such as SUPG, PSPG, and so on.

This paper is organized as follows. In Section 2, we briefly recall the velocity method and give the description of the shape minimization problem for Navier–Stokes flow. Section 3 is devoted to the computation of the discrete shape gradient of the discretized cost functional. Finally in Section 4, we propose a gradient type algorithm with some numerical examples. Conclusion follows.

2. Preliminaries and statement of the problem

2.1. Domain perturbation

We first recall the velocity method (see Cea [5] and Zolesio [8,9,32]) to generate domain perturbations for smooth domains Ω. The displacement of the point $X \in \Omega$ is governed by the differential system

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t, X) = \boldsymbol{V}\big(t, x(t)\big), \qquad x(0, X) = X,$$

which generates a transformation $T_t(\boldsymbol{V})$ defined as $T_t(X) = x(t)$. We denote the "transformed domain" $T_t(\boldsymbol{V})(\Omega)$ by $\Omega_t(\boldsymbol{V})$ at $t \geqslant 0$.

Let $J(\Omega)$ be a real-valued functional associated with any regular domain Ω, we say that the functional $J(\Omega)$ has a **Eulerian derivative** at Ω in the direction $\boldsymbol{V}$ if the limit

$$\lim_{t\searrow 0} \frac{1}{t}\big[J(\Omega_t) - J(\Omega)\big] := \mathrm{d}J(\Omega; \boldsymbol{V})$$

exists. Furthermore, if the map $\boldsymbol{V} \mapsto \mathrm{d}J(\Omega; \boldsymbol{V}) : C([0,\tau]; [\mathcal{D}^k(\mathbb{R}^N)]^N) \to \mathbb{R}$ is linear and continuous, we say that J is **shape differentiable** at Ω. In the distributional sense we have

$$\mathrm{d}J(\Omega; \boldsymbol{V}) = \langle \nabla J, \boldsymbol{V}\rangle_{(\mathcal{D}^k(\mathbb{R}^N)^N)' \times \mathcal{D}^k(\mathbb{R}^N)^N}. \tag{1}$$

When J has a Eulerian derivative, we say that ∇J is the **shape gradient** of J at Ω.

In the discrete level, in order to transport base functions on Ω onto the base functions on Ω_t, we should use a reference element for the definition of the transformation T_t. Denote by $\mathcal{T}_h$ a triangulation of Ω. Let K be any triangle with vertices a_i, $i = 1, 2, 3$. Let P_1 denote the set of all polynomials in two variables of degree $\leqslant 1$. There exists a unique invertible affine mapping

$$F_K : \hat{x} \in \mathbb{R}^2 \to F_K(\hat{x}) = B_K\hat{x} + b_K,$$

where B_K is an invertible 2×2 constant matrix and b_K a vector in $\mathbb{R}^2$. We then have

$$F_K(\hat{a}_i) = a_i, \quad i = 1, 2, 3,$$

where $\hat{a}_i$ are the vertices of the associated reference triangle element $\hat{K}$.

Now we can define the transformation T_t by using the reference element