Computers & Fluids 119 (2015) 101–114

Contents lists available at ScienceDirect

Computers & Fluids

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

A robust direct-forcing immersed boundary method with enhanced stability for moving body problems in curvilinear coordinates

L. Nicolaou [a], S.Y. Jung [a], T.A. Zaki [a,b,*]

[a] Department of Mechanical Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, UK
[b] Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

ARTICLE INFO

Article history:
Received 19 November 2014
Received in revised form 29 May 2015
Accepted 13 June 2015
Available online 2 July 2015

Keywords:
Immersed boundary method
Numerical stability
Curvilinear coordinates
Moving body

ABSTRACT

A robust immersed boundary method for semi-implicit discretizations of the Navier–Stokes equations on curvilinear grids is presented. No-slip conditions are enforced via momentum forcing, and mass conservation at the immersed boundary is satisfied via a mass source term developed for moving bodies. The errors associated with an explicit evaluation of the momentum forcing are analysed, and their influence on the stability of the underlying Navier–Stokes solver is examined. An iterative approach to compute the forcing term implicitly is proposed, which reduces the errors at the boundary and retains the stability guarantees of the original semi-implicit discretization of the Navier–Stokes equations. The implementation in generalized curvilinear coordinates and the treatment of moving boundaries are presented, followed by a number of test cases. The tests include stationary and moving boundaries and curvilinear grid problems (decaying vortex problem, stationary cylinder, flow in 90° bend in circular duct and oscillating cylinder in fluid at rest).

1. Introduction

Immersed boundary (IB) methods have become an established approach for modelling complex and moving geometries. The main advantage and the popularity of these methods are due to their simplicity and efficiency. Unlike body-conforming methods which require a body-fitted grid and remeshing in moving boundary problems, a structured Cartesian grid is adopted for the IB method. The effect of the body surface is included through the addition of boundary forces in the Navier–Stokes equations. Use of structured grids greatly simplifies the task of grid generation, particularly for moving bodies and leads to more efficient computational algorithms with better convergence and stability properties. The present work focuses on the use of IB in the context of semi-implicit Navier–Stokes solvers, which are commonly adopted in simulations of moderate- and high-Reynolds number flows.

IB methods can be grouped into continuous forcing and discrete forcing approaches [1]. The first immersed boundary method was developed by Peskin [2] and was applied to elastic boundaries moved by the fluid. Modifications to this approach for use with rigid boundaries were proposed by Beyer and Leveque [3] and Goldstein et al. [4] and employed *feedback forcing* to drive the velocity at the boundary to rest. However, these methods produced spurious oscillations and were subject to severe stability constraints. Another drawback of continuous forcing methods is the fact that a sharp representation of the boundary cannot be obtained since smoothing functions are used to transmit the forcing to the fluid, effectively spreading the location of the boundary. This is undesirable, especially when modelling high-Reynolds-number flows in which thin boundary layers need to be resolved accurately.

Mohd-Yusof [5] proposed a discrete derivation of the forcing term, in what is now commonly referred to as *direct forcing*. Other discrete forcing approaches exist, such as immersed interface methods (IIM) and Cartesian grid methods. However, the direct forcing approach remains most popular due to its simplicity and enhanced stability compared to other immersed boundary methods. Many variants of direct forcing methods have therefore appeared in the literature [6–13] and have been implemented in the framework of the fractional step algorithm which is commonly employed for solving the Navier–Stokes equations [7–9,14].

The accuracy and stability of direct forcing approaches used in conjunction with fractional step depend on the formulation of the fractional-step method, the computation of the forcing term and the treatment of mass conservation at the boundary. The two implementations of the fractional step method, referred to as the *p*-form, which neglects the pressure term in the

* Corresponding author at: Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA. Tel.: +1 (410)516 6599; fax: +1 (410)516 7254.
E-mail address: t.zaki@jhu.edu (T.A. Zaki).

http://dx.doi.org/10.1016/j.compfluid.2015.06.030
0045-7930/

intermediate velocity equation, and the Δp-form, which includes the pressure term from the previous time step, are both used extensively. However, only IB conditions applied in conjunction with the Δp-form are second-order accurate in time, whereas the p-form is only first-order accurate. The computation of the forcing term should be viewed relative to the temporal discretization of the governing equations. Explicit schemes for the solution of the Navier–Stokes equations are straightforward to implement, however they are limited by the viscous stability constraint. Therefore, most fractional step methods apply a semi-implicit approach where the diffusive terms are treated implicitly – these are the focus of the present work. In this context, implicit evaluation of the IB forcing term is not straightforward unless a simplified interpolation is adopted, for example the one used by Fadlun et al. [7]. Kim et al. [8] proposed an alternate approach where they provisionally advance the velocity field explicitly in order to compute the forcing term and then add it to the semi-implicit momentum equations. Their method has a clear advantage in terms of algorithmic efficiency. However, this approach can potentially reduce the stability limit of the numerical scheme due to the mismatch in the temporal discretization of the IB forcing and the governing equations.

In the present study, the errors in the computation of the forcing term are analysed and a stable second-order accurate direct forcing method is proposed. The current method consists of an iterative approach which decreases the errors at the boundary and enhances stability. The proposed method has been developed for use in a generalized curvilinear system allowing a wide range of complex geometries to be modelled efficiently on structured grids. The treatment of moving boundaries is also presented for completeness, and builds on the recent literature.

Sharp-interface IB methods are known to suffer from spurious force oscillations (SFOs) in moving body problems [15–20]. Lee et al. [18] identified two main sources of spurious oscillations: (i) The first source is the temporal discontinuity in the velocity which arises as a point from the fluid becomes solid and its velocity is suddenly changed to satisfy the no-slip condition at the IB. (ii) The second source is the spatial discontinuity in the pressure field which arises due to the momentum forcing and which contaminates the fluid field when a point from the solid becomes fluid. Seo and Mittal [19] found the major source of oscillations to be the violation of mass conservation near the immersed boundary. In order to suppress spurious oscillations, Yang and Balaras [16] proposed a field-extension approach in which the pressure and velocity at solid points becoming fluid were extrapolated from the surrounding fluid. Uhlmann [15] combined the direct forcing approach at Lagrangian points with discrete delta functions [2]. Lee et al. [18] showed that the addition of a mass source/sink inside the solid equally suppressed the SFOs, and Seo and Mittal [19] applied a cut-cell method to improve local mass conservation and reduce spurious oscillations. In the present method, an extension of the mass source term by Kim et al. [8] for use with moving boundaries is applied. Recently, a similar method applied to cells cut by the boundary was presented by Lee and You [20], and the differences will be discussed.

In summary, the stability of explicit and implicit forcing methods in the semi-implicit discretization of the Navier–Stokes equations is examined. An iterative implicit scheme is proposed, which is shown to have favourable stability properties. The method is capable of handling complex geometries on curvilinear grids and moving body problems. The paper is organized as follows: In Section 2, the governing equations and discretization scheme are presented. The accuracy of the IB boundary conditions in the fraction step method is discussed. In Section 3, the stability of explicit and implicit forcing methods is examined. An error analysis of explicit forcing methods is performed and the proposed implicit forcing approach is then presented. The stability of both methods is studied for flow over a stationary cylinder. The implementation of the immersed boundary conditions, the modifications required for extension onto curvilinear coordinates, and the treatment of moving boundaries are described in Section 4. In Section 5, numerical tests which validate the accuracy of the method are presented. Finally in Section 6, some conclusions are drawn.

2. Governing equations and semi-implicit discretization

In order to satisfy both the no-slip and no-penetration conditions at the immersed boundary, a momentum forcing, f_i, and a mass source term, q, are applied to the Navier–Stokes equations, similar to the method by Kim et al. [8]. The forcing term sets the velocity at points surrounding the boundary, which are referred to as IB points, to a particular value such that the velocity at the surface of the immersed body satisfies the boundary conditions. Cells containing the immersed boundary do not satisfy mass conservation without appropriate treatment. Therefore a mass source is added in order to ensure that mass is conserved [8].

The governing equations for unsteady incompressible flow are the momentum and continuity equations given below

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j x_j}+f_i, \tag{1}$$

$$\frac{\partial u_i}{\partial x_i}-q=0, \tag{2}$$

where x_i are the Cartesian coordinates, u_i are the corresponding velocities, p is the pressure, f_i are the momentum forcing components and q is the mass source term. The flow equations are solved on a staggered curvilinear grid using a volume flux formulation [21]. The equations are spatially discretized by a second-order finite-volume scheme and advanced in time with a second-order semi-implicit fractional step method that uses Adams–Bashforth for the convective terms and Crank–Nicolson for the diffusive terms. The flow solver has been extensively validated and adopted in direct numerical simulations of transitional and turbulent flows [22–24].

A number of approaches can be adopted for transformation of the governing equations from Cartesian to curvilinear coordinates, each with different methods of discretization, choice of dependent variables and grid layouts. For example, Cartesian velocities, contravariant velocities or volume fluxes, could be chosen as the dependent variables. While volume fluxes are used in our work, for generality and in keeping with the literature on IB methods, the discretized equations will be shown in Cartesian coordinates. Extension of the direct forcing method to curvilinear grids is independent of the coordinate transformation used and will be discussed in Section 4.1.

The fractional step method decouples the solution of the momentum equations (Eq. (1)) from that of the continuity equation (Eq. (2)) by solving them separately in two steps. An intermediate velocity field which is not divergence-free is computed first and subsequently corrected with a pseudo-pressure, ϕ, such that continuity is satisfied.

The discretized equations are given by

$$\frac{\hat{u}_i-u_i^{n-1}}{\Delta t}=\frac{1}{Re}\left(\alpha L(\hat{u}_i)+\beta L(u_i^{n-1})\right)-Gp^{n-1}-\gamma N(u_i^{n-1}) -\delta N(u_i^{n-2})+f_i^n, \tag{3}$$

$$DG\phi^n=\frac{1}{\Delta t}(D\hat{u}-q^n), \tag{4}$$

$$u_i^n=\hat{u}_i-\Delta t G\phi^n, \tag{5}$$

$$p^n=p^{n-1}+\phi^n, \tag{6}$$