Journal of Computational Physics 302 (2015) 59–67

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Short note

Computing jump conditions for the immersed interface method using triangular meshes

Sheng Xu *, Glen D. Pearson Jr.

Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156, USA

ARTICLE INFO

Article history:
Received 19 August 2014
Received in revised form 5 May 2015
Accepted 13 August 2015
Available online 2 September 2015

Keywords:
The immersed interface method
Triangular meshes
Jump conditions
Poisson solvers
Complex geometries
Non-smooth interfaces
Cartesian grid methods

ABSTRACT

The immersed interface method (IIM) can be employed to solve many interface problems on fixed Cartesian grids by incorporating necessary interface-induced *Cartesian* jump conditions into numerical schemes. In this paper, we present a method to compute the necessary Cartesian jump conditions from given *principal* jump conditions using triangular mesh representation of an interface. The triangular mesh representation is simpler and robuster than interface parametrization for a complex or non-smooth interface. We test our method by using the computed Cartesian jump conditions in the IIM to solve a Poisson equation subject to an interface with the shape of a sphere, cube, cylinder or cone. The results demonstrate the expected second-order accuracy of the solution in the infinity norm.

1. Introduction

The immersed interface method (IIM) was first proposed by LeVeque and Li [2] to solve elliptic equations with discontinuous coefficients and singular sources. Since then it has been developed to be a general methodology for solving various interface problems and irregular-domain problems [3].

The key idea of the IIM is to incorporate interface-induced jump conditions into discretization schemes to achieve desired solution accuracy and solving efficiency. To give an example on how jump conditions are incorporated into a numerical scheme, we consider the central finite difference approximation of a derivative of a discontinuous function. Let $g(s)$ be a function with jump discontinuities at $s = l$ and $s = r$ that fall within a three-node stencil as $s_{i-1} < l < s_i \le r < s_{i+1}$, where s_{i-1}, s_i and s_{i+1} are the coordinates of the stencil nodes satisfying $s_{i+1} - s_i = s_i - s_{i-1} = h$. The central finite difference scheme for $g''(s_i)$ on this stencil reads [5,6]

$$\frac{\mathrm{d}^2 g(s_i)}{\mathrm{d}s^2} = \frac{g(s_{i+1}) - 2g(s_i) + g(s_{i-1})}{h^2} + O(h^2) \\ - \frac{1}{h^2}\left(\sum_{n=0}^{3} \frac{-[g^{(n)}(l)]}{n!}(s_{i-1} - l)^n + \sum_{n=0}^{3} \frac{[g^{(n)}(r)]}{n!}(s_{i+1} - r)^n\right), \tag{1}$$

* Corresponding author.
E-mail address: sxu@smu.edu (S. Xu).
URL: http://faculty.smu.edu/sxu (S. Xu).

http://dx.doi.org/10.1016/j.jcp.2015.08.019
0021-9991/

where $[g^{(n)}(l)] = g^{(n)}(l^+) - g^{(n)}(l^-)$ denotes the jump conditions of the n-th derivatives of $g(s)$ at $s = l$, and similarly $[g^{(n)}(r)]$ the jump conditions at $s = r$. This example shows that we can modify a *standard* finite difference scheme to approximate derivatives of a discontinuous function by adding some extra terms that involve necessary jump conditions. In particular, a PDE for an interface problem can be discretized on a fixed Cartesian grid by incorporating into standard discretization schemes the jump conditions of the solution and its derivatives with respect to Cartesian coordinates. We name these jump conditions *Cartesian* jump conditions.

For the function p, we call

$$[p], \quad \left[\frac{\partial p}{\partial \mathbf{n}}\right], \quad [\Delta p]$$

the *principal* jump conditions, which are the jump conditions of p, the normal derivative of p and the Laplacian of p across an interface, respectively, where $\mathbf{n}$ denotes a unit normal vector for the interface, and a jump condition is hereafter defined as $[\cdot] = (\cdot)_{\mathbf{n}^+} - (\cdot)_{\mathbf{n}^-}$ with $\mathbf{n}^+$ and $\mathbf{n}^-$ denoting right and left limits respectively. These principal jump conditions can often be derived from the physics of a problem. The goal of this paper is to obtain from the given principal jump conditions the following Cartesian jump conditions

$$[p], \quad \left[\frac{\partial p}{\partial x_i}\right], \quad \left[\frac{\partial^2 p}{\partial x_i \partial x_j}\right]$$

where x_i and x_j ($i, j \in \{1, 2, 3\}$) are Cartesian coordinates in 3D.

In [6], a systematic approach based on interface parametrization was proposed to derive the Cartesian jump conditions from the principal jump conditions. However, interface parametrization is practically difficult for complex or non-smooth interfaces. In this paper, we propose a method to compute the Cartesian jump conditions using triangular mesh representation of an interface, which is more flexible and robust to deal with complex or non-smooth interfaces.

To test our method, we use the Cartesian jump conditions computed by the method in the IIM to solve the Poisson equation $\Delta p = f$ subject to the given principal jump conditions across a closed interface in 3D. Based on Eqn. (1), the Poisson equation can be discretized on a uniform Cartesian grid as

$$\Delta_h p_{ijk} = f_{ijk} + C_{ijk},$$

where Δ_h is the standard seven-point discrete Laplacian at the grid point (i, j, k), h is the spatial step of the grid, and C_{ijk} denotes the extra terms involving the necessary Cartesian jump conditions. The jump contribution C_{ijk} is nonzero only if the interface cuts through the seven-point Laplacian stencil at the grid point (i, j, k).

According to Eqn. (1), the local truncation error of the seven-point discrete Laplacian is $O(h)$ if the jump conditions of the third-order Cartesian derivatives are not included. However, it is difficult or costly to obtain the jump conditions of the third-order or higher Cartesian derivatives. Fortunately, second-order accuracy of the solution in the infinity norm can still be achieved in solving a Poisson equation subject to an interface if the local truncation error is reduced from $O(h^2)$ to $O(h)$ only for stencils cut by the interface [1]. So we only need to compute the Cartesian jump conditions listed above. Realizing that $(s_{i-1} - l)$ and $(s_{i-1} - r)$ in Eqn. (1) is at most $O(h)$, we need to compute $\left[\frac{\partial p}{\partial x_i}\right]$ with $O(h^2)$ accuracy and $\left[\frac{\partial^2 p}{\partial x_i \partial x_j}\right]$ with $O(h)$ accuracy.

One motivation of the work in this paper is to solve a fluid flow around moving rigid objects on a fixed Cartesian grid using the IIM with boundary condition capturing [8,10]. The flow is formulated such that the rigid objects are replaced by the virtual fluid enclosed by singular forces concentrating on the surfaces of the objects via the Dirac delta function, which induce jump discontinuities in the velocity and the pressure across the interfaces between the virtual and real fluids. The principal jump conditions for the velocity and the pressure are related with the singular forces [6], which can be determined explicitly in the boundary condition capturing approach [8–10]. Using the method proposed in this paper, we can obtain the Cartesian jump conditions explicitly. They appear at the right-hand sides of linear systems resulting from the discretization of the Navier–Stokes equations without modifying the coefficient matrices corresponding to the background standard numerical schemes, so standard fast solvers such as FFT solvers can be employed to solve the linear systems. Linnick and Fasel [4] presented a high-order IIM in which Cartesian jump conditions are directly computed from unknown solutions using one-sided finite difference schemes. Since their Cartesian jump conditions are solution-dependent, the coefficient matrices of their linear systems are modified and iterative solvers are required.

2. Cartesian jump conditions

Suppose that a triangular mesh representation of a closed interface is given. See Fig. 1 for examples. Suppose that the edge lengths of the triangular mesh is of $O(h)$. Suppose that the principal jump conditions $[p]$, $\left[\frac{\partial p}{\partial \mathbf{n}}\right]$, and $[\Delta p]$ are given at appropriate locations (specified later) on the triangular mesh. The Cartesian jump condition $[p]$ is given as a principal jump condition. We compute the Cartesian jump conditions $\left[\frac{\partial p}{\partial x_i}\right]$ and $\left[\frac{\partial^2 p}{\partial x_i \partial x_j}\right]$ at each vertex of the triangular mesh.