Commun. Comput. Phys.
doi: 10.4208/cicp.201210.250212a

Vol. **12**, No. 5, pp. 1588-1602
November 2012

High Order Cubic-Polynomial Interpolation Schemes on Triangular Meshes

Renzhong Feng*

School of Mathematics and Systematic Science & Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beijing University of Aeronautics and Astronautics, Beijing 100191, China.

Received 20 December 2010; Accepted (in revised version) 25 February 2012

Available online 11 June 2012

Abstract. The Cubic-Polynomial Interpolation scheme has been developed and applied to many practical simulations. However, it seems the existing Cubic-Polynomial Interpolation scheme are restricted to uniform rectangular meshes. Consequently, this scheme has some limitations to problems in irregular domains. This paper will extend the Cubic-Polynomial Interpolation scheme to triangular meshes by using some spline interpolation techniques. Numerical examples are provided to demonstrate the accuracy of the proposed schemes.

AMS subject classifications: 65M99, 82D37, 82-08

Key words: Cubic-Polynomial Interpolation scheme, hyperbolic equations, triangular mesh.

1 Introduction

The compact cubic interpolated propagation (CIP) scheme (see, e.g., [1,2]) is based on the cubic-polynomial interpolation ideas, which was developed for solving general hyperbolic equations and is found of low diffusion and good stability properties. In the method a spatial profile within each grid is interpolated with a cubic polynomial, and both the values f and its spatial derivative ∇f on the grid are predicted in advance. The first derivative in the CIP scheme is calculated from a model equation for the spatial derivative which is consistent with the master equation. This scheme has been successfully applied to various complex fluid flow problems [3] and wave propagation problem [4], and has been extended to conservative form [5] and body-fitted grid system [6].

At present the CIP method is mainly designed for rectangle or quadrilateral meshes. It is known that triangular meshes have advantages for irregular domains and it is natural to extend the CIP method to the triangular meshes. This will be main purpose of this

*Corresponding author. *Email address:* `fengrz@buaa.edu.cn` (R. Feng)

 ©2012 Global-Science Press

work. Designing higher order schemes with irregular meshes satisfying certain properties for hyperbolic conservation laws has been an important research target in recent years, see, e.g., [7–10].

The layout of this paper is as follows. The basic idea is to use the Hermite interpolation techniques for triangles, which will be described in the next section. In section 3, we present the CIP scheme on triangular mesh restricted to two space dimensions. Numerical examples will be presented in the final section.

2 Hermite interpolation and high-order differences

2.1 Hermite interpolation

We first introduce two different Hermite interpolation methods on triangles. Consider $\triangle_{123}$ as illustrated in Fig. 1 and let O be its barycentric point, $(x_j,y_j), 1\le j\le 3$ be the Cartesian coordinates of its three vertexes. For any point (x,y) inside the triangle, let f_1,f_2,f_3 and f_0 be the values of a smooth function $f(x,y)$ at the three vertexes and its barycentric point, respectively. Moreover, let $(f_x)_1,(f_x)_2,(f_x)_3$ be the three x-directional partial derivatives and $(f_y)_1,(f_y)_2,(f_y)_3$ be three y-directional partial derivative values.

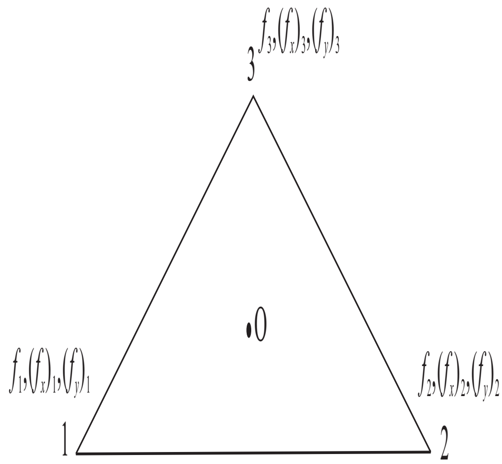

Figure 1: An illustrating triangle and some relevant values.

2.1.1 Method I

Consider the Hermite interpolation on $\triangle_{123}$:

$$H_1(f;x,y)=\sum_{i=1}^{3}\Big(\alpha_i^{(1)}(x,y)f_i+\beta_i^{(1)}(x,y)(f_x)_i+\gamma_i^{(1)}(x,y)(f_y)_i\Big), \tag{2.1}$$

where the coefficients for f_i are

$$\alpha_1^{(1)}(x,y)=l_1^3+3l_2l_1^2+3l_3l_1^2+2l_1l_2l_3, \tag{2.2a}$$
$$\alpha_2^{(1)}(x,y)=l_2^3+3l_1l_2^2+3l_3l_2^2+2l_1l_2l_3, \tag{2.2b}$$
$$\alpha_3^{(1)}(x,y)=l_3^3+3l_1l_3^2+3l_2l_3^2+2l_1l_2l_3, \tag{2.2c}$$

the coefficients for $(f_x)_i$ are

$$\beta_1^{(1)}(x,y)=(x_2-x_1)\Big(l_1^2l_2+\frac{1}{2}l_1l_2l_3\Big)+(x_3-x_1)\Big(l_1^2l_3+\frac{1}{2}l_1l_2l_3\Big), \tag{2.3a}$$
$$\beta_2^{(1)}(x,y)=(x_1-x_2)\Big(l_2^2l_1+\frac{1}{2}l_1l_2l_3\Big)+(x_3-x_2)\Big(l_2^2l_3+\frac{1}{2}l_1l_2l_3\Big), \tag{2.3b}$$
$$\beta_3^{(1)}(x,y)=(x_1-x_3)\Big(l_3^2l_1+\frac{1}{2}l_1l_2l_3\Big)+(x_2-x_3)\Big(l_3^2l_2+\frac{1}{2}l_1l_2l_3\Big), \tag{2.3c}$$

and the coefficients for $(f_y)_i$ are

$$\gamma_1^{(1)}(x,y)=(y_2-y_1)\Big(l_1^2l_2+\frac{1}{2}l_1l_2l_3\Big)+(y_3-y_1)\Big(l_1^2l_3+\frac{1}{2}l_1l_2l_3\Big), \tag{2.4a}$$
$$\gamma_2^{(1)}(x,y)=(y_1-y_2)\Big(l_2^2l_1+\frac{1}{2}l_1l_2l_3\Big)+(y_3-y_2)\Big(l_2^2l_3+\frac{1}{2}l_1l_2l_3\Big), \tag{2.4b}$$
$$\gamma_3^{(1)}(x,y)=(y_1-y_3)\Big(l_3^2l_1+\frac{1}{2}l_1l_2l_3\Big)+(y_2-y_3)\Big(l_3^2l_2+\frac{1}{2}l_1l_2l_3\Big). \tag{2.4c}$$

Here l_1,l_2,l_3 are the area coordinates of point (x,y) on $\triangle_{123}$, which are given by

$$l_1=\frac{1}{2s}\Big((x_2y_3-x_3y_2)+(y_2-y_3)x+(x_3-x_2)y\Big), \tag{2.5a}$$
$$l_2=\frac{1}{2s}\Big((x_3y_1-x_1y_3)+(y_3-y_1)x+(x_1-x_3)y\Big), \tag{2.5b}$$
$$l_3=\frac{1}{2s}\Big((x_1y_2-x_2y_1)+(y_1-y_2)x+(x_2-x_1)y\Big). \tag{2.5c}$$

The following relations hold:

$$l_1+l_2+l_3=1, \quad l_1,l_2,l_3\ge 0, \tag{2.6a}$$
$$x=x_1l_1+x_2l_2+x_3l_3, \quad y=y_1l_1+y_2l_2+y_3l_3, \tag{2.6b}$$

where s is the area of $\triangle_{123}$.

Our Method I is to require the following interpolation conditions:

$$H_1(f;x_i,y_i)=f_i, \quad \frac{\partial H_1}{\partial x}(x_i,y_i)=(f_x)_i, \quad \frac{\partial H_1}{\partial y}(x_i,y_i)=(f_y)_i, \quad i=1,2,3. \tag{2.7}$$

We point out that the Hermite interpolation H_1 can only reproduce the bi-variable polynomials of degree not exceeding 2. If we require that a Hermite interpolation on triangle can reproduce the polynomials of degree 3, then the Hermite interpolation must satisfy the following interpolation conditions

$$H_2(f;x_o,y_o)=f_o, \tag{2.8a}$$

$$H_2(f;x_i,y_i)=f_i, \quad \frac{\partial H_2}{\partial x}(x_i,y_i)=(f_x)_i, \quad \frac{\partial H_2}{\partial y}(x_i,y_i)=(f_y)_i, \quad i=1,2,3. \tag{2.8b}$$

2.1.2 Method II

Consider the Hermite interpolation satisfying the interpolation conditions (2.8b):

$$H_2(f;x,y)=\alpha_0^{(2)}(x,y)f_0+\sum_{i=1}^{3}\Big(\alpha_i^{(2)}(x,y)f_i+\beta_i^{(2)}(x,y)(f_x)_i+\gamma_i^{(2)}(x,y)(f_y)_i\Big), \tag{2.9}$$

where $\alpha_0^{(2)}(x,y)=27l_1l_2l_3$, the coefficients for f_i are given by

$$\alpha_1^{(2)}(x,y)=l_1^3+3l_1^2(l_2+l_3)-7l_1l_2l_3, \tag{2.10a}$$

$$\alpha_2^{(2)}(x,y)=l_2^3+3l_2^2(l_1+l_3)-7l_1l_2l_3, \tag{2.10b}$$

$$\alpha_3^{(2)}(x,y)=l_3^3+3l_3^2(l_1+l_2)-7l_1l_2l_3, \tag{2.10c}$$

the coefficients for $(f_x)_i$ are given by

$$\beta_1^{(2)}(x,y)=(x_2-x_1)(l_1^2l_2-l_1l_2l_3)+(x_3-x_1)(l_1^2l_3-l_1l_2l_3), \tag{2.11a}$$

$$\beta_2^{(2)}(x,y)=(x_3-x_2)(l_2^2l_3-l_1l_2l_3)+(x_1-x_2)(l_2^2l_1-l_1l_2l_3), \tag{2.11b}$$

$$\beta_3^{(2)}(x,y)=(x_1-x_3)(l_3^2l_1-l_1l_2l_3)+(x_2-x_3)(l_3^2l_2-l_1l_2l_3), \tag{2.11c}$$

and the coefficients for $(f_y)_i$ are

$$\gamma_1^{(2)}(x,y)=(y_2-y_1)(l_1^2l_2-l_1l_2l_3)+(y_3-y_1)(l_1^2l_3-l_1l_2l_3), \tag{2.12a}$$

$$\gamma_2^{(2)}(x,y)=(y_3-y_2)(l_2^2l_3-l_1l_2l_3)+(y_1-y_2)(l_2^2l_1-l_1l_2l_3), \tag{2.12b}$$

$$\gamma_3^{(2)}(x,y)=(y_1-y_3)(l_3^2l_1-l_1l_2l_3)+(y_2-y_3)(l_3^2l_2-l_1l_2l_3). \tag{2.12c}$$

2.2 High-order differences on triangular meshes

In this subsection we will introduce a high-order difference method on triangular mesh to approximate with up to second-order accuracy; for more details see [11, 12].

Definition 2.1. Suppose $\mathcal{F}=\{f|f:\mathbb{R}^d\to\mathbb{R}\}$, A is a discrete subset of $\mathbb{R}^d$, $\alpha=(\alpha_1,\cdots,\alpha_d)\in Z_+^d$, $D^\alpha=D^{\alpha_1}\cdots D^{\alpha_d}$ is the derivative with order $|\alpha|=\alpha_1+\cdots+\alpha_d$, $\mathcal{P}_n=\mathcal{P}_n(\mathbb{R}^d)$ is the set of multivariate polynomials of degree $\le n$. An operator $\mathcal{D}_A^\alpha:\mathcal{F}\to\mathcal{F}$ is said to be a $\mathcal{P}_n$-exact A-discretization of $\mathcal{D}^\alpha$ if and only if

(a) There exists a real vector $\lambda=(\lambda_a)_{a\in A}$ such that, for any $f\in\mathcal{F}$ and $X\in\mathbb{R}^d$,

$$(\mathcal{D}_A^\alpha f)(X)=\sum_{a\in A}\lambda_a f(X+a); \tag{2.13}$$

(b) For any $p\in\mathcal{P}_n$,

$$\mathcal{D}_A^\alpha p=\mathcal{D}^\alpha p. \tag{2.14}$$

Then we also say that $\mathcal{D}_A^\alpha f$ is a $\mathcal{P}_n$-exact A-discretization of $\mathcal{D}^\alpha f$. If the points in set A are properly posed for $\mathcal{P}_n$, then $\mathcal{D}_A^\alpha$ is determined uniquely.

Suppose the point set $A=\{Q_1(x_1,y_1),\cdots,Q_6(x_6,y_6)\}$ is posed in $\mathcal{P}_2=\mathcal{P}_2(\mathbb{R}^2)$, i.e., there does not exist a nonzero polynomial in $\mathcal{P}_2$ which vanishes on all the points of A, where $\mathcal{P}_2$ is the bi-variate polynomial space of polynomials of degree not exceeding 2. The formulas for computing the divided difference $f[A]^{(1,0)}$ defined in [11] is given by

$$f[A]^{(1,0)}=\frac{1}{(1,0)!}\mathcal{D}_A^{(1,0)}f(0)=\sum_{i=1}^{6}\lambda_i f(Q_i), \tag{2.15}$$

where $(\alpha_1,\alpha_2)!=\alpha_1!\alpha_2!$ and the coefficients $\{\lambda_i\}_{i=1}^6$ are determined by

$$M_1\begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\\ \lambda_5\\ \lambda_6\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix}, \tag{2.16a}$$

where

$$M_1:=\begin{pmatrix}1&1&1&1&1&1\\ x_1&x_2&x_3&x_4&x_5&x_6\\ y_1&y_2&y_3&y_4&y_5&y_6\\ x_1y_1&x_2y_2&x_3y_3&x_4y_4&x_5y_5&x_6y_6\\ x_1^2&x_2^2&x_3^2&x_4^2&x_5^2&x_6^2\\ y_1^2&y_2^2&y_3^2&y_4^2&y_5^2&y_6^2\end{pmatrix}. \tag{2.16b}$$

Therefore, we have

$$\mathcal{D}_A^{(1,0)}f(X)=\sum_{i=1}^{6}\lambda_i f(Q_i+X),\quad X=(x,y)\in\mathbb{R}^2. \tag{2.17}$$

According to (2.15) and (2.16a)-(2.16b), the computing formula of divided difference $f[A_X]^{(1,0)}$ on six points $A_X=\{Q_1-X,Q_2-X,Q_3-X,Q_4-X,Q_5-X,Q_6-X\}$ is given by

$$f[A_X]^{(1,0)}=\frac{1}{(1,0)!}\mathcal{D}_{A_X}^{(1,0)}f(0)=\sum_{i=1}^{6}\lambda_{i,X}f(Q_i-X), \tag{2.18}$$

where coefficients $\{\lambda_{i,X}\}_{i=1}^{6}$ are determined by the following equations

$$M_2\begin{pmatrix}\lambda_{1,X}\\ \lambda_{2,X}\\ \lambda_{3,X}\\ \lambda_{4,X}\\ \lambda_{5,X}\\ \lambda_{6,X}\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix}, \tag{2.19a}$$

where

$$M_2:=\begin{pmatrix}1&1&1&1&1&1\\ x_1-x&x_2-x&x_3-x&x_4-x&x_5-x&x_6-x\\ y_1-y&y_2-y&y_3-y&y_4-y&y_5-y&y_6-y\\ xy_{11}&xy_{22}&xy_{33}&xy_{44}&xy_{55}&xy_{66}\\ (x_1-x)^2&(x_2-x)^2&(x_3-x)^2&(x_4-x)^2&(x_5-x)^2&(x_6-x)^2\\ (y_1-y)^2&(y_2-y)^2&(y_3-y)^2&(y_4-y)^2&(y_5-y)^2&(y_6-y)^2\end{pmatrix}, \tag{2.19b}$$

with $xy_{jj}=(x_j-x)(y_j-y)$. Therefore, we have

$$D_{A_X}^{(1,0)}f(X)=\sum_{i=1}^{6}\lambda_{i,X}f(Q_i). \tag{2.20}$$

Similarly, on the point set $A=\{Q_1,Q_2,Q_3,Q_4,Q_5,Q_6\}$, the computing formula of divided difference $f[A]^{(0,1)}$ defined in [11] is given by

$$f[A]^{(0,1)}=\frac{1}{(0,1)!}D_A^{(0,1)}f(0)=\sum_{i=1}^{6}\eta_i f(Q_i), \tag{2.21}$$

where coefficients $\{\eta_i\}_{i=1}^{6}$ are determined by the following equations

$$M_1\begin{pmatrix}\eta_1\\ \eta_2\\ \eta_3\\ \eta_4\\ \eta_5\\ \eta_6\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\\0\\0\end{pmatrix}, \tag{2.22}$$

where M_1 is given by (2.16b). Therefore, we have

$$\mathcal{D}_A^{(0,1)}f(X)=\sum_{i=1}^{6}\eta_i f(Q_i+X). \tag{2.23}$$

According to (2.21) and (2.22), the computing formula of divided difference $f[A_X]^{(0,1)}$ on six points $A_X=\{Q_1-X,Q_2-X,Q_3-X,Q_4-X,Q_5-X,Q_6-X\}$ is given by

$$f[A_X]^{(0,1)}=\frac{1}{(0,1)!}\mathcal{D}_{A_X}^{(0,1)}f(0)=\sum_{i=1}^{6}\eta_{i,X}f(Q_i-X), \tag{2.24}$$

where coefficients $\{\eta_{i,X}\}_{i=1}^{6}$ are determined by

$$M_2\begin{pmatrix}\eta_{1,X}\\ \eta_{2,X}\\ \eta_{3,X}\\ \eta_{4,X}\\ \eta_{5,X}\\ \eta_{6,X}\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\\0\\0\end{pmatrix}, \tag{2.25}$$

where M_2 is defined by (2.19b). Therefore, we have

$$\mathcal{D}_{A_X}^{(0,1)}f(X)=\sum_{i=1}^{6}\eta_{i,X}f(Q_i). \tag{2.26}$$

Similarly, the computing formula of divided difference $f[A_X]^{(2,0)}$ on six points $A_X=\{Q_1-X,Q_2-X,Q_3-X,Q_4-X,Q_5-X,Q_6-X\}$ is

$$f[A_X]^{(2,0)}=\frac{1}{(2,0)!}\mathcal{D}_{A_X}^{(2,0)}f(0)=\sum_{i=1}^{6}\kappa_{i,X}f(Q_i-X). \tag{2.27}$$

Consequently, we have

$$\mathcal{D}_{A_X}^{(2,0)}f(X)=2\sum_{i=1}^{6}\kappa_{i,X}f(Q_i). \tag{2.28}$$

The computing formula of divided difference $f[A_X]^{(0,2)}$ on six points $A_X=\{Q_1-X,Q_2-X,Q_3-X,Q_4-X,Q_5-X,Q_6-X\}$ is

$$f[A_X]^{(0,2)}=\frac{1}{(0,2)!}\mathcal{D}_{A_X}^{(0,2)}f(0)=\sum_{i=1}^{6}\tau_{i,X}f(Q_i-X). \tag{2.29}$$

Consequently, we have

$$\mathcal{D}_{A_X}^{(0,2)}f(X)=2\sum_{i=1}^{6}\tau_{i,X}f(Q_i). \tag{2.30}$$

The coefficients $\{\kappa_{i,X}\}_{i=1}^{6}$ in (2.28) and $\{\tau_{i,X}\}_{i=1}^{6}$ in (2.30) are determined by

$$M_2\begin{pmatrix}\kappa_{1,X}\\ \kappa_{2,X}\\ \kappa_{3,X}\\ \kappa_{4,X}\\ \kappa_{5,X}\\ \kappa_{6,X}\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\1\\0\end{pmatrix},\quad M_2\begin{pmatrix}\tau_{1,X}\\ \tau_{2,X}\\ \tau_{3,X}\\ \tau_{4,X}\\ \tau_{5,X}\\ \tau_{6,X}\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\\0\\1\end{pmatrix},\tag{2.31}$$

respectively. It follows from [11] and Definition 2.1 that

$$\mathcal{D}^{(1,0)}_{A_X}f(X)\approx\frac{\partial f}{\partial x}(X),\qquad \mathcal{D}^{(0,1)}_{A_X}f(X)\approx\frac{\partial f}{\partial y}(X),\tag{2.32a}$$

$$\mathcal{D}^{(2,0)}_{A_X}f(X)\approx\frac{\partial^2 f}{\partial x^2}(X),\quad \mathcal{D}^{(0,2)}_{A_X}f(X)\approx\frac{\partial^2 f}{\partial y^2}(X),\quad X\in\mathbb{R}^2,\tag{2.32b}$$

and when $f(x,y)=1,x,y,x^2,xy,y^2$, it can be verified that

$$\mathcal{D}^{(1,0)}_{A_X}f(X)=\frac{\partial f}{\partial x}(X),\qquad \mathcal{D}^{(0,1)}_{A_X}f(X)=\frac{\partial f}{\partial y}(X),\tag{2.33a}$$

$$\mathcal{D}^{(2,0)}_{A_X}f(X)=\frac{\partial^2 f}{\partial x^2}(X),\quad \mathcal{D}^{(0,2)}_{A_X}f(X)=\frac{\partial^2 f}{\partial y^2}(X).\tag{2.33b}$$

3 The CIP scheme on triangular meshes

Let us consider a two-dimensional hyperbolic equation in Cartesian coordinates x,y:

$$L_2 f\equiv\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}=g.\tag{3.1}$$

The idea of the CIP method for solving Eq. (3.1) on a triangle mesh is similar to that on a rectangle mesh [1, 2]. Suppose that the solution domain of Eq. (3.1) is partitioned by a triangle mesh and point (x_i,y_i) is a grid point on the triangle mesh, which adjoins m points $\{(x_{j_k},y_{j_k})\}_{k=1}^{m}$ (see Fig. 2). From the idea of the CIP method, we first judge which triangle with vertex (x_i,y_i) point $(x_i-u*\Delta t,y_i-v*\Delta t)$ locates on. It is well known that if $(x_i-u*\Delta t,y_i-v*\Delta t)$ locates on the triangle $\triangle_{i,j_k,j_{k+1}}$, then the area coordinates l_1,l_2,l_3 corresponding to point $(x_i-u*\Delta t,y_i-v*\Delta t)$ are all non-negative; otherwise there exists at least one negative value among l_1,l_2,l_3. If the point $(x_i-u*\Delta t,y_i-v*\Delta t)$ locates on $\triangle_{i,j_k,j_{k+1}}$, then we interpolate $f(x,y)$ within $\triangle_{i,j_k,j_{k+1}}$ by Method I of (2.1) or Method II of (2.8b), i.e.,

$$H_1(f;x,y)=\sum_{l=i,j_k,j_{k+1}}\Big(\alpha_l^{(1)}(x,y)f_l+\beta_l^{(1)}(x,y)(f_x)_l+\gamma_l^{(1)}(x,y)(f_y)_l\Big);\tag{3.2a}$$

$$H_2(f;x,y)=\alpha_0^{(2)}(x,y)f_o+\sum_{l=i,j_k,j_{k+1}}\Big(\alpha_l^{(2)}(x,y)f_l+\beta_l^{(2)}(x,y)(f_x)_l+\gamma_l^{(2)}(x,y)(f_y)_l\Big).\tag{3.2b}$$

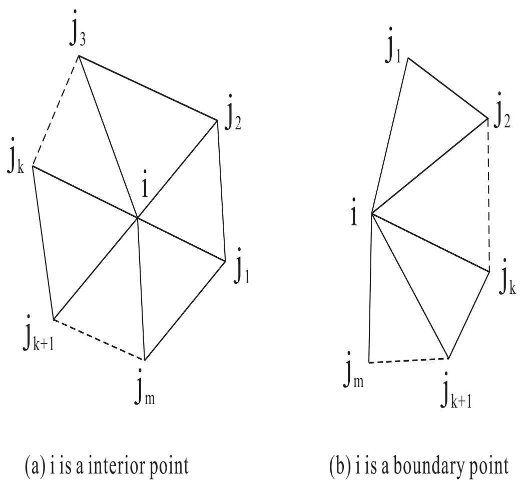

Figure 2: Point i and its neighboring points on a triangle mesh.

If the values for f at all grid points and the barycentric points of all triangles, i.e., f_i and f_{o_i} $(1 \le i \le i_{max})$, are known, only six parameters $(f_x)_i, (f_x)_{j_k}, (f_x)_{j_{k+1}}, (f_y)_i, (f_y)_{j_k}, (f_y)_{j_{k+1}}$ remain to be determined. From the idea of the CIP method, see, e.g., [1,2], we know that the first spatial derivative must be determined consistently with the master equation. Thus we have obtained quite accurate solutions without solving a matrix equation to determine the polynomial. We will use similar approach here. The equations for the first spatial derivatives are derived from Eq. (3.1), namely, from $(L_2 f - g)_x = 0$ and $(L_2 f - g)_y = 0$:

$$L_2(f_x) = g_x - \frac{\partial u}{\partial x}\frac{\partial f}{\partial x} - \frac{\partial v}{\partial x}\frac{\partial f}{\partial y} \equiv R_x, \tag{3.3a}$$

$$L_2(f_y) = g_y - \frac{\partial u}{\partial y}\frac{\partial f}{\partial x} - \frac{\partial v}{\partial y}\frac{\partial f}{\partial y} \equiv R_y. \tag{3.3b}$$

If f, f_x and f_y at all grid points and f_o at barycentric point of each triangle are given by Eqs. (3.1), (3.3a) and (3.3b), then we can calculate the values for $H_1(f;x,y)$ in Eq. (3.2a) and $H_2(f;x,y)$ in (3.2b) at any point (x,y) on $\triangle_{i,j_k,j_{k+1}}$.

Following [1,2], we split Eqs. (3.1), (3.3a) and (3.3b) into two phases:

(a) the non-advection phase

$$\frac{\partial f}{\partial t} = g, \tag{3.4a}$$

$$\frac{\partial f_x}{\partial t} = g_x - \frac{\partial u}{\partial x}\frac{\partial f}{\partial x} - \frac{\partial v}{\partial x}\frac{\partial f}{\partial y}, \tag{3.4b}$$

$$\frac{\partial f_y}{\partial t} = g_y - \frac{\partial u}{\partial y}\frac{\partial f}{\partial x} - \frac{\partial v}{\partial y}\frac{\partial f}{\partial y}; \tag{3.4c}$$

(b) the advection phase

$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}=0, \tag{3.5a}$$

$$\frac{\partial f_x}{\partial t}+u\frac{\partial f_x}{\partial x}+v\frac{\partial f_x}{\partial y}=0, \tag{3.5b}$$

$$\frac{\partial f_y}{\partial t}+u\frac{\partial f_y}{\partial x}+v\frac{\partial f_y}{\partial y}=0. \tag{3.5c}$$

The quantities f,f_x and f_y in the non-advection phase (3.4) are advanced according to

$$f_i^*=f_i^n+g_i^n\Delta t, \tag{3.6a}$$

$$(f_x)_i^*=(f_x)_i^n+((g_x)_i^n-(u_x)_i(f_x)_i^n-(v_x)_i(f_y)_i^n)\Delta t, \tag{3.6b}$$

$$(f_y)_i^*=(f_y)_i^n+((g_y)_i^n-(u_y)_i(f_x)_i^n-(v_y)_i(f_y)_i^n)\Delta t. \tag{3.6c}$$

In (3.6), the quantities $g_i^n,(g_x)_i^n,(g_y)_i^n,(u_x)_i,(u_y)_i,(v_x)_i$ and $(v_y)_i$ may be solved using high-order difference methods (2.19a)-(2.19b) and (2.28)-(2.30). We also adopt the way in [2] to approximate $(g_x)_i^n$ and $(g_y)_i^n$

$$(g_x)_i^n\sim\frac{(f_i^*)_x-(f_x)_i^n}{\Delta t},\quad (g_y)_i^n\sim\frac{(f_i^*)_y-(f_y)_i^n}{\Delta t}, \tag{3.7}$$

where $(f_i^*)_x$ and $(f_i^*)_y$ are difference values obtained by using formulas (2.19a)-(2.19b) and (2.26) with values $\{f_i^*\}$.

After the non-advection phase is solved, the CIP method is applied to advection phase. The solution for the equation $L_2h=0$, where h is f,f_x or f_y, after a very short time Δt can be estimated as

$$h(x,y,t+\Delta t)\sim h(x-u\Delta t,y-v\Delta t,t). \tag{3.8}$$

Thus, the profile after one time step Δt for Method I is

$$f_i^{n+1}=H_1(f;x_i-u\Delta t,y_i-v\Delta t),\quad (f_x)_i^{n+1}=\Big(\frac{\partial H_1}{\partial x}\Big)_i,\quad (f_y)_i^{n+1}=\Big(\frac{\partial H_1}{\partial y}\Big)_i; \tag{3.9}$$

and for Method II is

$$f_i^{n+1}=H_2(f;x_i-u\Delta t,y_i-v\Delta t),\quad (f_x)_i^{n+1}=\Big(\frac{\partial H_2}{\partial x}\Big)_i,\quad (f_y)_i^{n+1}=\Big(\frac{\partial H_2}{\partial y}\Big)_i. \tag{3.10}$$

The above can be written in an explicit way:

$$f_i^{n+1}=\sum_{l=i,j_k,j_{k+1}}\Big(\alpha_l^{(1)}(x,y)f_l^*+\beta_l^{(1)}(x,y)(f_x)_l^*+\gamma_l^{(1)}(x,y)(f_y)_l^*\Big),\quad \text{Method I}, \tag{3.11a}$$

$$\begin{aligned}f_i^{n+1}=\alpha_0^{(2)}(x,y)(f_o)_i^*+\sum_{l=i,j_k,j_{k+1}}\Big(&\alpha_l^{(2)}(x,y)f_l^*+\beta_l^{(2)}(x,y)(f_x)_l^*\\&+\gamma_l^{(2)}(x,y)(f_y)_l^*\Big),\quad \text{Method II},\end{aligned} \tag{3.11b}$$

where $\triangle_{i,j_k,j_{k+1}}$ is the triangle which contains point $(x_i-u\Delta t,y_i-v\Delta t)$.

4 Numerical examples

In what follow, we present some numerical results on regular triangle meshes and general triangle meshes to illustrate the performance of the methods.

4.1 Constant coefficient linear advection equation

Example 4.1. We consider following two-dimensional constant coefficient linear advection problem

$$\frac{\partial f}{\partial t}+a\frac{\partial f}{\partial x}+b\frac{\partial f}{\partial y}=0, \quad (x,y)\in[-1,1]\times[-1,1], \quad t>0, \tag{4.1}$$

where a,b are the propagation velocities of wave along x,y directions respectively and constants. We take the initial solution as $f(x,y,0)=\alpha\exp(-\beta x^2-\gamma y^2)$, with some constants α,β,γ.

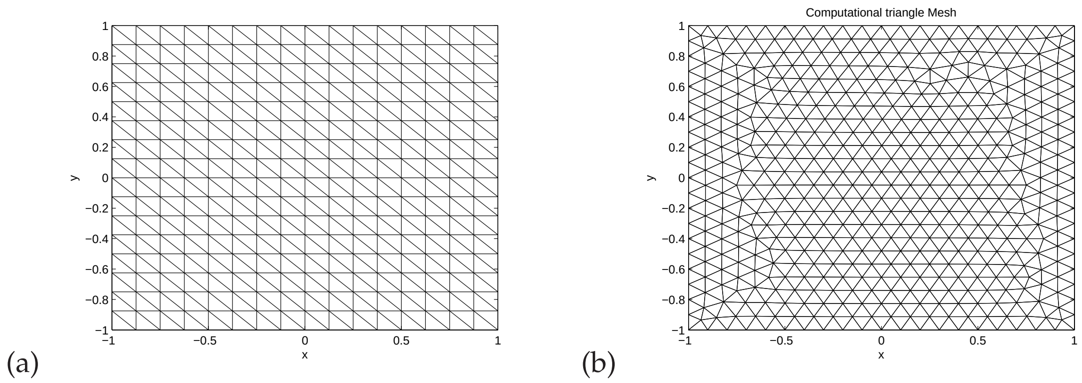

Figure 3: Meshes used in the numerical test: a regular triangular mesh (a) and a general triangular mesh (b).

Take $a=b=0.5,\alpha=1,\beta=\gamma=8$. The exact solution of this problem is $f(x,y,t)=\alpha\exp(-\beta(x-at)^2-\gamma(y-bt)^2)$. We first examine the performance of the methods on regular triangle meshes, see Fig. 3(a). Table 1 shows the computational errors and convergence orders for Methods I and II after 50 steps under the CFD condition number $c=0.1$.

It is observed from Table 1 that Method I obtains third-order convergence in time and space on regular triangle meshes, while Method II yields a 4th-order.

Next Methods I and II will be implemented on general triangle meshes, i.e., generated by standard routines, see Fig. 3(b). Table 2 also shows the computational errors and convergence orders for Methods I and II on some general triangle meshes, for Method I we used 50 steps with $\Delta t=0.03$, and for Method II we used 205 steps with $\Delta t=0.01$. From the results of Table 2 we can see that Method I obtains a convergence order close to 3 in space on general triangle mesh, while Method II has a 4th-order. Compared with the results on regular triangle meshes, the convergence orders with the general triangular mesh drop a bit.

Table 1: Example 4.1: the computational error and convergence orders on regular meshes.

	Method I		Method II	
$N_x \times N_y$	L_∞ error	order	L_∞ error	order
10×10	1.07e-01		5.24e-02	
20×20	1.25e-02	3.10	4.80e-03	3.45
40×40	1.40e-03	3.16	3.19e-04	3.91
80×80	1.68e-04	3.06	2.02e-05	3.98
160×160	2.06e-05	3.03	1.27e-06	4.00

Table 2: Same as Table 1, except on a general triangular mesh.

	Method I			Method II		
Number of elements	L_∞ error	Δt	order	L_∞ error	Δt	order
240	5.76e-02	0.03		4.90e-02	0.01	
928	9.60e-03	0.03	2.65	5.70e-03	0.01	3.18
2228	2.90e-03	0.03	2.73	1.20e-03	0.01	3.56
3690	1.40e-03	0.03	2.87	4.47e-04	0.01	3.90

4.2 Variable coefficient linear advection equation

Example 4.2. Consider the following two-dimensional variable coefficient linear advection problem

$$\frac{\partial f}{\partial t} - y\frac{\partial f}{\partial x} + x\frac{\partial f}{\partial y} = 0, \quad (x,y)\in[-1,1]\times[-1,1], \quad t>0, \tag{4.2}$$

where $(-y,x)$ is the propagation velocity of wave. We also take the initial solution as $f(x,y,0)=\alpha\exp(-\beta x^2-\gamma y^2)$, where the constants are taken as $\alpha=1,\beta=4,\gamma=12$.

The exact solution of Eq. (4.2) is

$$f(x,y,t)=\alpha\exp\Big(-(\beta(x^2+y^2)\cos^2(\theta(x,y)-t)+\gamma(x^2+y^2)\sin^2(\theta(x,y)-t))\Big), \tag{4.3}$$

where

$$\theta(x,y)=\begin{cases} \arctan(y/x), & x>0, \\ \pi+\arctan(y/x), & x<0. \end{cases} \tag{4.4}$$

Similarly to Section 4.1, we first examine the numerical performance on the regular triangle mesh (see Fig. 3(a)). The solution process is again divided into two phases, with the first one being the non-advection phase:

$$\frac{\partial f_x}{\partial t} = -\frac{\partial f}{\partial y}, \quad \frac{\partial f_y}{\partial t} = \frac{\partial f}{\partial x}, \tag{4.5}$$

Table 3: Example 4.2: the computational results for Methods I and II on regular meshes.

	Method I		Method II	
$N_x \times N_y$	L_∞ error	order	L_∞ error	order
10×10	3.20e-03		2.00e-03	
20×20	3.30e-04	3.28	1.71e-4	3.55
40×40	3.91e-05	3.08	1.16e-05	3.88
80×80	4.84e-06	3.02	7.43e-07	3.96
160×160	6.01e-07	3.01	4.95e-08	3.90

Table 4: Same as Table 3, except on a general mesh.

	Method I			Method II		
Number of elements	L_∞ error	$\triangle$t	order	L_∞ error	$\triangle$t	order
240	4.14e-02	0.0008		3.14e-02	0.0008	
928	8.40e-03	0.0008	2.36	3.80e-03	0.0008	3.13
2228	2.80e-03	0.0008	2.51	7.83e-04	0.0008	3.61
3690	1.40e-03	0.0008	2.73	3.07e-04	0.0008	3.70

and the second one being the advection phase:

$$\frac{\partial f}{\partial t} - y\frac{\partial f}{\partial x} + x\frac{\partial f}{\partial y} = 0, \tag{4.6a}$$

$$\frac{\partial f_x}{\partial t} - y\frac{\partial f_x}{\partial x} + x\frac{\partial f_x}{\partial y} = 0, \tag{4.6b}$$

$$\frac{\partial f_y}{\partial t} - y\frac{\partial f_y}{\partial x} + x\frac{\partial f_y}{\partial y} = 0. \tag{4.6c}$$

Table 3 shows the computational results for Methods I and II after 1000 steps under the CFL conditions $c = 0.001$.

It is observed from Table 3 that Method I again gives the 3rd-order of convergence, while Method II yields a 4th-order of convergence.

Next we examine the numerical performance on the general triangular mesh, as given by Fig. 3(b). Table 4 shows the computational errors and convergence orders for Methods I and II after 2750 steps under Δt=0.0008. The choice of small time step allows us to check the order of convergence in space. Again it is seen from Table 4 that Method II gives one order higher, but both methods have about half-order drop in the corresponding orders.

4.3 Linear advection-diffusion problem

Example 4.3. Finally we consider the two-dimensional linear advection-diffusion problem

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} = \varepsilon_x \frac{\partial^2 f}{\partial x^2} + \varepsilon_y \frac{\partial^2 f}{\partial x^2}, \quad 0 \le x, y \le 1, \quad t > 0. \tag{4.7}$$

when $\varepsilon_x = \varepsilon_y = 0.05$. The initial condition is $f(x,y,0) = 1$, and the boundary conditions are

$$f(x,0,t) = e^{20x-40}, \quad f(x,1,t) = e^{20x-20}, \quad 0<x<1, \quad t>0, \tag{4.8a}$$

$$f(1,y,t) = e^{20y-20}, \quad f(0,y,t) = e^{20y-40}, \quad 0<y<1, \quad t>0. \tag{4.8b}$$

The steady solution of the above problem is

$$f(x,y) = e^{20(x+y-2)}, \quad 0\le x,y\le 1. \tag{4.9}$$

The calculation process is divided into two phases: The first one is non-advection phase:

$$\frac{\partial f}{\partial t} = \varepsilon_x \frac{\partial^2 f}{\partial x^2} + \varepsilon_y \frac{\partial^2 f}{\partial y^2}, \tag{4.10a}$$

$$\frac{\partial f_x}{\partial t} = \varepsilon_x \left(\frac{\partial^2 f}{\partial x^2}\right)_x + \varepsilon_y \left(\frac{\partial^2 f}{\partial y^2}\right)_x, \tag{4.10b}$$

$$\frac{\partial f_y}{\partial t} = \varepsilon_x \left(\frac{\partial^2 f}{\partial x^2}\right)_x + \varepsilon_y \left(\frac{\partial^2 f}{\partial y^2}\right)_y, \tag{4.10c}$$

and the second one is advection phase:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} = 0, \tag{4.11a}$$

$$\frac{\partial f_x}{\partial t} + \frac{\partial f_x}{\partial x} + \frac{\partial f_x}{\partial y} = 0, \tag{4.11b}$$

$$\frac{\partial f_y}{\partial t} + \frac{\partial f_y}{\partial x} + \frac{\partial f_y}{\partial y} = 0. \tag{4.11c}$$

Table 5 shows L_∞ errors at steady state under two different calculation methods, from which it is observed that Method I has the comparable ability as P^2-DG, i.e., the DG method with polynomial space P^2.

Table 5: Example 4.3: the L_∞ error comparison with P^2-DG and Method I.

$N_x \times N_y$	Polynomial space P^2 for DG		CIP method I	
	L_∞ error	Order	L_∞ error	Order
10 × 10	1.25e-01		4.87e-02	
20 × 20	3.22e-02	1.96	2.08e-02	1.23
40 × 40	5.99e-03	2.43	8.10e-03	1.36
80 × 80	9.21e-04	2.70	1.0e-03	3.02

5 Conclusion

In summary, we extended the cubic-polynomial interpolation method designed for rectangular meshed to triangular meshes. Two high-order schemes, one is of order 3 and another is of order 4 in space, are proposed, which are also tested for accuracy verification.

Some preliminary tests show that the present extension is useful for solving convection-diffusion type problems on general triangular meshes. Since the algorithms are developed in a compact explicit form, it seems that they are ideally suited for adaptivity and parallelizability. The application of these methods to nonlinear hyperbolic equations or systems remain to be our future investigations.

Acknowledgments

Part of this work was finished during the author's visit to Department of Mathematics at Hong Kong Baptist University. The author would like to thank Prof. Tao Tang and Dr. Leevan Ling for many useful discussions, and the editors for their help in English writing. The research is partially supported by the National Natural Science Key Foundation of China No. 10931004 and ISTCP of China grant No. 2010DFR00700.

References

[1] T. Yabe and E. Takei, A new higher-order Godunov method for general hyperbolic equations, J. Phys. Soc. Japan, 57(8) (1988), 2598-2601.

[2] T. Yabe, T. Ishikawa, P. Y. Wang, T. Aoki and Y. Kadota, Cubic-polynomial interpolation. II. Two-and three-dimensional solvers, Comput. Phys. Commun., 66 (1991), 233-242.

[3] T. Yabe, F. Xiao and T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys., 169 (2001), 556-593.

[4] K. Shiraishi and T. Matsuoka, Wave propagation simulation using the CIP method of characteristic equations, Commun. Comput. Phys., 3(1) (2008), 121-135.

[5] T. Nakamura, R. Tanaka, T. Yabe and K. Takizawa, Exactly conservative semi-Lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique, J. Comput. Phys., 174 (2001), 171-207.

[6] T. Yabe, H. Mizoe, K. Takizawa, H. Moriki, H.-N. Im and Y. Ogata, Higher-order schemes with CIP method and adapitve Soroban grid towards mesh-free scheme, J. Comput. Phys., 194 (2004), 57-77.

[7] Y. Liu, Central schemes on overlapping cells, J. Comput. Phys., 209 (2005), 82-104.

[8] Y. Liu, C.-W. Shu, E. Tadmor and M. Zhang, Non-oscillatory hierarchical reconstruction for central and finite volume schemes, Commun. Comput. Phys., 2 (2007), 933-963.

[9] G. Hu, R. Li and T. Tang, A robust WENO type finite volume solver for steady Euler equations on unstructured grids, Commun. Comput. Phys., 9 (2011), 627-648.

[10] S. Mishra and E. Tadmor, Constraint preserving schemes using potential-based fluxes. I. Multidimensional transport equations, Commun. Comput. Phys., 9 (2011), 688-710.

[11] C. Rabut, Multivariate divided difference with simple knots, SIAM J. Numer. Anal., 38(4) (2001), 1294-1311.

[12] R. Feng and X. Zhou, A kind of multiquadric quasi-interpolation opertors satisfying any degree polynomial reproduction property to scattered data, J. Comput. Appl. Math., 235 (2011), 1502-1514.