# Interactive PDE patch-based surface modeling from vertex-frames

**Shuangbu Wang**[1] · **Yu Xia**[1] · **Lihua You**[1] · **Hassan Ugail**[2] · **Alfonso Carriazo**[3] · **Andres Iglesias**[4] · **Jianjun Zhang**[1] ·

**Abstract** Polygon, subdivision, and NURBS are three mainstream modeling techniques widely applied in commercial software packages. They require heavy manual operations, and involve a lot of design variables leading to big data, high storage costs and slow network transmissions. In this paper, we integrate the strengths of boundary-based surface creation and partial differential equation (PDE) based geometric modeling to obtain the first analytical $C^0$ continuous 4-sided PDE patches involving sculpting force-based shape creation and manipulation and use them to develop an interactive modeling technique for easy and quick creation of 3D models with small data from vertex-frames. With this modeling technique, a vertex frame is defined by eight vertices, and a $C^0$ continuous 4-sided PDE patch is created from the vertex-frame through an analytical solution to a vector-valued second-order PDE subjected to the boundary conditions determined by the eight vertices of a vertex-frame. A user-friendly interface is developed from the obtained analytical solution, which enables users to interactively input and modify vertex-frame models easily and create 3D models in real time. Different surface modeling tasks are carried out to test the developed interactive tool and compare our proposed method with polygon and NURBS modeling and Coons surfaces. The results demonstrate the effectiveness of our proposed method and its advantages in reducing design variables, saving storage costs, and effective shape creation and manipulation.

✉Shuangbu Wang

E-mail: swang1@bournemouth.ac.uk

[1] National Centre for Computer Animation, Bournemouth University, Bournemouth BH12 5BB, UK.
[2] The Centre for Visual Computing, University of Bradford, Bradford BD7 1DP, UK.
[3] Department of Geometry and Topology, Faculty of Mathematics, University of Seville, Apdo. Correos 1160, 41080 Seville, Spain.
[4] Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros s/n, E-39005 Santander, Spain.

# 1 Introduction

Surface modeling is widely applied in creative and industrial sectors to produce 3D geometric models. Current mainstream surface modeling techniques are polygon [1], Non-uniform rational B-splines (NURBS) [2] and subdivision [3, 4]. These modeling techniques have been integrated into high-end geometric modeling systems such as Maya, Houdini and 3D Max. All of the three mainstream modeling approaches are purely geometric, which means that no extra constraint forces called sculpting forces are used in creation and manipulation of surface shapes.

Polygon and subdivision modeling techniques are surface vertex-based. They use planar facets to approximate curved surfaces of 3D objects. In order to represent curved surfaces and create detailed 3D models, a large number of planar facets defined by vertices (design variables) are required, which cause big data, high storage costs, slow network transmissions and heavy manual operations to manipulate the vertices. NURBS modeling is control point-based. It also involves many design variables and requires extra manual operations to stitch adjacent NURBS surfaces together with required continuities and add control points to interactively manipulate NURBS surfaces in small deformation regions as shown in Fig. 7 of this paper in comparison with PDE-based modeling. Boundary-based surface creation generates surfaces from boundary curves or boundary curves and tangents (normal derivative functions on boundary curves). Since only boundary curves or boundary curves and tangents are required to define surfaces, fewer design variables and manual operations are involved. The representative boundary-based surface creation methods are bilinear Coons surfaces and bicubic Coons surfaces [5]. Unfortunately, both of them have weak ability in creating and manipulating shapes of surfaces. Bilinear Coons surfaces are defined by four boundary curves and have no ability to change their shapes if boundary curves remain unchanged. Bicubic Coons surfaces are defined by four boundary curves and four normal derivative functions on the four boundary curves. Shapes of bicubic Coons surfaces can be changed through the four normal derivative functions. Such a shape manipulation method is less capable. First, when the normal derivative functions on the boundary curves of one bicubic Coons patch are changed, the normal derivative functions on the shared boundary curves of all adjacent patches must be changed accordingly to maintain the required continuity. Second, adjusting the normal derivative functions on the boundary curves of one bicubic Coons patch changes the shape of the whole patch only and cannot change the shape in arbitrarily specified local regions in the patch as shown in Fig. 8 of this paper. Third, many shape changes that can be obtained by manipulating control points or surface vertices or sculpting forces cannot be achieved by changing the normal

derivative functions on the boundary curves of bicubic Coons surfaces, which is also demonstrated in Fig. 8 of this paper.

Partial differential equations (PDEs) were introduced into geometric modeling about three decades ago to develop PDE-based geometric modeling. Before that, they had already been applied in engineering and scientific calculations to describe the underlying physics. For example, a fourth-order PDE can be used to describe the underlying physics of bending deformations of a thin elastic plate subjected to lateral loads, and a wave equation, which is a second-order PDE, can be used to describe the underlying physics of transverse vibrations of a tensed string [6]. Due to this nature, PDE-based geometric modeling is physics-based, has a potential to create more realistic appearances of 3D models and 3D deformations, and addresses the weaknesses of purely geometric modeling.

PDE-based modeling uses the solution to a vector-valued PDE subjected to user's specified boundary conditions to create a PDE patch. Like the term on the right-hand side of a fourth-order PDE describing bending deformations of a thin elastic plate, which is a lateral force acting on the plate, the term on the right-hand side of an arbitrary vector-valued PDE used for geometric modeling acts as a sculpting force. It and the coefficients of all the terms on the left-hand side of a vector-valued PDE called shape control parameters greatly affect the shape of a PDE patch and provide flexible and powerful shape manipulation handles. As shown in Fig. 11 of Subsection 5.1, different shape control parameters and different values of same shape control parameters can be used to create different shapes of a same PDE patch. A sculpting force can be applied not only in both a whole PDE patch and small regions encircled by boundary curves in a single PDE patch shown in Fig. 8 in Subsection 4.2, but also a large region across multiple PDE patches shown in Fig. 9 in the same subsection.

Since PDE surface patches are defined by the solution to a PDE subjected to exact satisfaction of boundary conditions, adjacent PDE patches automatically achieve required continuities on shared boundaries defined in boundary conditions. Unlike polygon modelling, which uses many planar facets to approximate curved surfaces causing a lot of design variables and cannot achieve tangent and higher order continuities between adjacent planar facets, PDE-based modeling creates curved patches with complicated shapes directly, greatly reduces design variables, and easily achieves tangent and higher order continuities between adjacent PDE patches. Different from NURBS modelling, which involves many control points, and requires extra manual operations to stitch adjacent NURBS patches together and add control points to interactively create shapes in small deformation regions, PDE-based modeling only requires boundary information and few coefficients in a vector-valued PDE to define complicated 3D models leading to fewer design variables, naturally achieves specified continuities without any manual operations to stitch two PDE patches together, and uses a sculpting force to create various shapes in any deformation regions more easily and flexibly. Same as bicubic Coons surfaces, PDE-based modeling can also use the normal derivative functions on the boundary curves of PDE patches to

manipulate PDE surfaces. But differently, PDE-based modeling has following advantages as demonstrated in Fig. 8 of this paper. First, it uses shape control parameters and a sculpting force to change the shape of a PDE patch without changing the continuities on the boundary curves shared by adjacent PDE patches. Second, it applies a sculpting force to achieve shape manipulation in arbitrarily specified local regions in a PDE patch. Third, it easily creates various shape changes by applying different sculpting forces and adjusting shape control parameters. Therefore, PDE-based modeling integrates Coons surface's strengths of easiness and few design variables, but overcomes the weakness of Coons surfaces in shape creation and manipulations.

However, due to the difficulty in analytically solving PDEs, existing tools of manipulating sculpting forces are mainly developed from an analytical solution to a vector-valued PDE for simple 2-sided PDE patches [7] or from a numerical solution for complicated 3D models [8]. In this paper, we will propose an analytical solution to a PDE involving a sculpting force and use it to create and manipulate 4-sided patches.

PDE-based geometric modeling has been intensively studied. Various numerical methods such as the finite element method [9, 10, 11], finite difference method [12, 13], and direct discretization of polygonal models using a discrete Laplace operator [14, 15] etc. have been proposed to solve PDEs for hybrid subdivision surface design [16], surface reconstruction [15], estimation of surface normal [14], smoothing arbitrary triangle meshes [17], surface deformations [18], defining geometric solid models [19], surface modeling [20, 9], and direct shape manipulation [8] etc.

When using PDE-based modeling to create new 3D models from scratch, 3D models are decomposed into some 2-, 3-, and 4-sided PDE patches. For creating 3D models from PDE patches, numerical solutions of PDEs have three weaknesses. 1) The analytical functions of the boundary constraints for a 3- or 4-sided surface patch cannot be exactly satisfied since numerical methods obtain the solutions at discrete points. 2) Numerical methods such as the finite element and finite difference methods involve heavy computations, and may not be ideal in interactive geometric modeling or real-time shape manipulations. 3) Numerical methods are needed to obtain coordinate values at many discrete nodes or vertices. Each node or vertex is a vector-valued design variable consisting of 3 components. Therefore, numerical methods will lead to many design variables and high storage costs.

The above discussions highlight the biggest problem for PDE-based geometric modeling. It is how to develop analytical solutions for PDE patches and interactive user interface from analytical PDE patches for releasing the potential of PDE-based geometric modeling.

Since analytical solutions of 3- and 4-sided PDE patches are very difficult to obtain, the existing research studies on analytical PDE surfaces focus on 2-sided PDE patches, i. e., creating a PDE patch from boundary conditions on two opposite boundaries. Various accurate and approximate analytical solutions of 2-sided PDE patches have been developed [21, 7, 22, 23, 24, 25, 26, 27, 28, 29, 30]. Although the paper [30] claims an analytical pseudo-spectral method, which

can generate regular 4-sided PDE surface patches, no analytical formulae were presented to explicitly determine the unknown constants. In addition, the paper only used the proposed pseudo-spectral method to create single 4-sided PDE patch, how to use the proposed 4-sided PDE patches to create 3D models was not investigated in the paper. In [31], a central point and its surrounding three curves were used to develop an analytical solution of PDE surfaces and significantly reduce the size of a polygon femur model and the corresponding NURBS model from 3.2 MB (polygon model) and 0.55 MB (NURBS model) to 0.26 MB (PDE model). The method is effective in rebuilding polygon models with a much smaller data size, but unapplicable to creation of 3D models from scratch. In addition, 4-sided PDE patches were not investigated in the paper. We are unaware of any other work provides analytical solution for 4-side PDE patches.

Using analytical 2-sided PDE patches only is not applicable to all 3D modeling tasks. For example, 2-sided PDE patches are incapable in creating branched models. Here, a branched model is a 3D model with branching structures that are connected to but not part of the central body of the model, such as the trunk of a tree with branches and a human body with limbs. In order to release the potential of PDE-based geometric modeling, analytical 3- and 4-sided patches with different continuity requirements and powerful shape manipulation functions should be developed. Due to the limit of space, in this paper, we will propose the first analytical $C^0$ continuous 4-sided PDE patches, investigate how to integrate shape control parameters and sculpting forces in the analytical $C^0$ continuous 4-sided PDE patches, and develop an interactive interface to facilitate 3D model creation and manipulation from the analytical $C^0$ continuous 4-sided PDE patches. The development of analytical $C^n$ continuous 3- and 4-sided PDE patches and demonstration of sculpting forces in creating the deformations in the regions bounded by complicated boundary curves will be our following work.

The contributions of this paper are: 1) obtaining the first analytical $C^0$ continuous 4-sided PDE patches to create complicated 3D models and release the potential of PDE-based geometric modeling, 2) integrating shape control parameters and sculpting forces into the analytical $C^0$ continuous 4-sided PDE patches to achieve flexible and powerful shape creation and deformations and overcome the weakness of Coons surfaces in shape creation and manipulations, 3) proposing vertex-frames consisting of eight boundary vertices to represent boundary conditions for PDE patches, reduce design variables and data size of PDE-based geometric modeling, and facilitate the development of the analytical $C^0$ continuous 4-sided PDE patches, 4) investigating a vertex frame-based modeling technique to quickly create 3D models with positional continuity by filling vertex-frames with the analytical 4-sided PDE patches, and 5) developing a user-friendly interface, which enables users to interactively input and modify vertex-frames, create 3D surface models, and manipulate shapes of the created 3D models.

The remaining parts of this paper are organized as follows. The related works on purely geometric modeling approaches and PDE-based modeling

methods are briefly reviewed in Section 2. Analytical $C^0$ continuous 4-sided PDE patches are investigated in Section 3. The results and comparisons with polygon, NURBS, and Coons surfaces are given in Section 4. Interactive design of 3D modeling is developed in Section 5. And finally, conclusions and future work are discussed in Section 6.


## 2 Related work

The work presented in this paper is related to purely geometric modeling techniques and PDE-based geometric modeling. In this section, we first briefly introduce polygon modeling, subdivision, and patch-based modeling in relation to the proposed method, and then review existing work on PDE-based geometric modeling.


### 2.1 Purely geometric modeling techniques

Purely geometric modeling techniques include polygon, subdivision, and patch surfaces such as NURBS and Coons surfaces etc. Among them, polygon, subdivision, and NURBS have become industrial standard being integrated in many commercial software packages.

Polygon modeling [1] can produce detailed or branched models, assign *uv* texture coordinates, and create hard edges more readily than NURBS modeling. However, polygons are incapable of accurately representing curved surfaces. Therefore, a large number of polygons must be generated to approximate curved surfaces in a visually appealing manner, leading to heavy manual operations, a large number of design variables, high storage costs, and slow network transmissions.

Subdivision modeling [32, 33, 34, 35] starts the modeling with a coarse polygonal model, subdivides its polygonal faces into smaller faces through approximating or interpolating schemes, and generates a denser polygon mesh of the model. Subdivision makes the modeling of complex geometry more easily and rendering more efficiently, but has the same weaknesses as polygon modeling.

The typical patch surfaces are NURBS surfaces [2, 36]. Patch surface-based modeling starts from a single NURBS patch and obtains the whole model by manually stitching many patches together [37] to deal with the continuity problem between different patches, leading to heavy manual operations. In addition, this modeling method is control point-based. Although it has fewer design variables than polygon modeling and subdivision, many control points are still required to create complicated 3D models.

Unlike NURBS surfaces, which are control point-based, Coons surfaces are boundary-based. A bilinear Coons surface [38] is constructed from four boundary curves, and bicubic Coons surfaces [39] is constructed from four boundary curves and four normal derivative functions on the four boundary

curves. Coons surfaces are easy to use, and involve fewer design variables than NURBS surfaces since they only involve boundary information. However, bilinear Coons surfaces are not adjustable if boundary curves are not changed. Although the shapes of bicubic Coons surfaces can be manipulated by changing the normal derivative functions on boundary curves, such shape manipulations are not flexible and powerful.

Polygon, NURBS and subdivision involve heavy manual operations and large data. Coons surfaces have small data but are weak in shape manipulation. The PDE-based modeling method will develop analytical 4-sided PDE patches and vertex frame-based 3D model creation to reduce large data and heavy manual operations, and introduce shape control parameters and sculpting forces to obtain flexible and powerful shape manipulations.

## 2.2 PDE-based geometric modeling

PDEs were introduced in geometric modeling by Bloor and Wilson about three decades ago [25]. After that, PDE-based geometric modeling attracts a lot of research attention. Various numerical, accurate analytical and approximate analytical solutions have been developed to promote their applications. Since the biggest problem for PDE-based geometric modeling is how to solve PDEs, we briefly review existing numerical methods, accurate analytical methods, and approximate analytical methods used to solve PDEs for geometric modeling applications.

### 2.2.1 Numerical methods

Numerical methods are most effective in solving PDEs for geometric modeling applications. Popular numerical methods are the finite element method, finite difference method, and direct discretization of polygonal models using a discrete Laplace-Beltrami operator. Since there are many publications on numerical PDE-based geometric modeling, it is impossible to review all of them. In what follows, only some of them are reviewed.

A B-spline finite element method was proposed in [10] and used to approximate PDE surfaces. A bivariant B-spline finite element method is developed in [9] and applied to tackle dynamic PDE surface modeling. By considering Laplacian and Laplacian gradient energies leading to biharmonic and triharmonic equations, respectively, and viewing the discretization of biharmonic and triharmonic equations as transformation of a mixed element discretization, a mixed finite element method was investigated in [11] to solve the biharmonic and triharmonic equations and address variational surface modeling. Through presenting a novel technique to evaluate the finite element basis functions and coupling the finite element method with a hybrid loop and Catmull-Clark subdivision algorithm, a numerical simulation method was examined in [16] for hybrid subdivision surface design using geometric PDEs.

By solving an Euler-Lagrange equation with the finite difference method, the smoothing properties were achieved by reducing the amplitude of various high frequency Fourier harmonics in surfaces [40]. Using the finite difference method to discretize a general sixth order geometric PDE, the problems of surface processing and modeling including creation of high order continuous surfaces are efficiently solved in [12]. Employing the finite difference discretization and variational interpolating approach with the localized iterative solver, an implicit fourth-order PDE and an implicit second-order PDE were numerically integrated in [13] to achieve shape design of solid models.

Using the umbrella operator to linearly approximate the Laplacian operator at each vertex of polygon models and numerically integrating a diffusion equation involving the Laplacian and the second Laplacian, rough features from irregularly data are rapidly removed to portray a smooth surface [41]. In order to smooth arbitrary triangle meshes while satisfying $G^1$ boundary conditions, a new algorithm was presented in [17] to numerically solve a PDE with the discrete Laplace-Beltrami operator and 1-neighborhood based discretization of the mean curvature normal at a vertex. The discrete Laplace-Beltrami operator was also used to obtain variational minimization for surface deformations in [18], estimate surface normal in [14], and perform surface reconstruction in [15].

### 2.2.2 Accurate analytical methods

Accurate analytical methods exactly satisfy both PDEs and boundary conditions. It is very difficult or even impossible to obtain accurate analytical solutions of PDEs when boundary conditions are complicated. Therefore, the existing work only developed accurate analytical solutions for 2-sided PDE patches subjected to some simple boundary conditions. Treating blending design as a boundary value problem and considering some simple boundary functions, the accurate analytical solutions for $x$ and $y$ components were obtained from a second-order PDE and the accurate analytical solution for $z$ component was obtained from a fourth-order PDE to create parametric surfaces [42]. For primary surfaces represented with a combination of hyperbolic sine and cosine functions, some accurate analytical solutions were presented to blend the primary surfaces together in [25]. By mapping a unit cube in $(u, v, w)$ parameter space to a hexahedral solid in physical space, the accurate analytical solutions to a second-order PDE with three parametric variables were obtained in [43] to investigate the functionality in solids. The accurate closed form solutions for the boundary conditions represented with triangular functions are investigated in [7] for vase design.

### 2.2.3 Approximate analytical methods

Approximate analytical solutions cannot exactly satisfy both PDEs and boundary conditions. Existing approximate analytical solutions exactly satisfy boundary conditions, but only minimize the error of PDEs. Fourier series-based

approximate analytical solution was proposed in [21] to generate free-form surfaces for the design of hull of yacht-like boat, propeller blade, phone handset, and ship hull. The Fourier series-based approximate analytical solution was used to solve a fourth-order PDE for the interactive surface design in [23] and parametric design of aircraft geometry in [24]. It was also extended to obtain an approximate analytical solution to a sixth-order PDE for surface modeling in [28]. A weighted residual method was presented to approximately solve a time-dependent fourth-order PDE for creation of deformable moving surfaces in [27], a time-independent fourth-order PDE in [26] for $C^1$ continuous surface blending, and a time-independent sixth-order PDE for surface modeling in [22] and $C^2$ continuous surface blending in [29].

Numerical methods are most effective in solving various PDEs but with discrete representations, big data, and high computational costs. Accurate analytical solutions are only applicable to some simple surface modeling tasks. And existing approximate analytical solutions can deal with more complicated surface modeling than accurate analytical solutions, but most studies focus on 2-sided PDE patches. How to develop explicit representations of analytical 4-sided PDE patches has not been well investigated.

## 3 Continuous and sculpting force-driven 4-sided PDE patches

In this section, we develop the first analytical 4-sided PDE patches, which not only maintain $C^0$ continuity on curved boundaries but also involve shape control parameters and sculpting forces to achieve flexible and powerful shape manipulations. In what follows, we first introduce the mathematical model of the 4-sided PDE patches that consists of a vector-valued second-order nonhomogeneous PDE and $C^0$ continuous boundary conditions in Subsection 3.1. Since the general solution of a vector-valued second-order nonhomogeneous PDE consists of a general solution of the corresponding homogeneous PDE and a particular solution satisfying the nonhomogeneous PDE, we will investigate the general solution of the corresponding homogeneous PDE in Subsection 3.2, and the particular solution in Subsection 3.3.

### 3.1 PDE-based mathematical model

A vector-valued second-order PDE is easier to solve analytically than higher order PDEs. It provides enough degrees of freedom to satisfy the conditions of $C^0$ continuity between adjacent patches. In order for the 4-sided PDE patches to have flexible and powerful shape manipulation functions, we also introduce a vector-valued sculpting force function to the right-hand side of the PDE. Based on these considerations, the PDE used to develop analytical 4-sided PDE surfaces takes the form of

$$\mathbf{a_1}\frac{\partial^2 \mathbf{S}(u,v)}{\partial u^2} + \mathbf{a_2}\frac{\partial^2 \mathbf{S}(u,v)}{\partial v^2} = \mathbf{F}(u,v) \tag{1}$$

where $\mathbf{S}(u,v) = [x(u,v) \ y(u,v) \ z(u,v)]^T$ represents a 4-sided PDE surface patch, $\mathbf{a_1}$ and $\mathbf{a_2}$ are vector-valued shape control parameters, $u$ and $v$ are the parametric variables defined by $0 \leq u \leq 1$ and $0 \leq v \leq 1$, and $\mathbf{F}(u,v) = [f_x(u,v) \ f_y(u,v) \ f_z(u,v)]^T$ are vector-valued sculpting forces.

A 4-sided patch $\mathbf{S}(u,v)$ has four boundaries. If the boundary curves on the four boundaries are: $\boldsymbol{\alpha}(u)$,$\boldsymbol{\beta}(u)$, $\boldsymbol{\gamma}(v)$, and $\boldsymbol{\delta}(v)$, respectively, the 4-sided patch $\mathbf{S}(u,v)$ at its four boundaries $u=0$, $u=1$, $v=0$, and $v=1$ should have the same curve functions as the four boundary curves, which leads to the following boundary conditions.

$$\mathbf{S}(u,0) = \boldsymbol{\alpha}(u), \mathbf{S}(u,1) = \boldsymbol{\beta}(u) \ \forall u \in [0,1]$$
$$\mathbf{S}(0,v) = \boldsymbol{\gamma}(v), \mathbf{S}(1,v) = \boldsymbol{\delta}(v) \ \forall v \in [0,1]$$

$$(2)$$

The mathematical model of our proposed 4-sided patches is defined by the vector-valued second-order PDE (1) and the $C^0$ continuous boundary conditions (2). The remaining work is how to derive the analytical solution of the mathematical model.

3.2 General solution of second-order homogeneous PDE

When deriving the general solution $\mathbf{S}_h(u,v)$ of the homogeneous PDE of Eq. (1), the right-hand side term is zero. We obtain

$$\mathbf{a_1}\frac{\partial^2 \mathbf{S}_h(u,v)}{\partial u^2} + \mathbf{a_2}\frac{\partial^2 \mathbf{S}_h(u,v)}{\partial v^2} = 0 \qquad (3)$$

The surface function of the 4-sided patches should exactly satisfy the positional continuous boundary conditions (2), have some degrees of freedom to minimize the error of the PDE (3), and introduce shape control parameters to manipulate the shape of 4-sided patches. Taking these into account, the vector-valued surface function of the 4-sided patches is taken to be

$$\mathbf{S}_h(u,v) = C^{(0)} + C^{(1)}u + C^{(2)}u^2 + C^{(3)}u^3 + C^{(4)}v +$$
$$C^{(5)}v^2 + C^{(6)}v^3 + C^{(7)}uv + C^{(8)}u^2v + C^{(9)}uv^2$$

$$(4)$$

where $C^{(i)}$ $(i = 0, 1, ..., 9)$ are the vector-valued unknown constants to be explicitly determined below.

We introduce vertex-frames to define four boundary curves for reducing design variables. As shown in Fig. 1(a), a vertex-frame consists of eight vertices: four corner vertices $P^{(1)}$, $P^{(3)}$, $P^{(5)}$, $P^{(7)}$, and four middle vertices $P^{(2)}$, $P^{(4)}$, $P^{(6)}$, $P^{(8)}$, and each of the four boundary curves is defined by three vertices of a vertex-frame. According to Eq. (2), the four corner vertices and the four middle vertices are related to the four boundary curves through $P^{(1)} = \boldsymbol{\alpha}(0)$, $P^{(2)} = \boldsymbol{\alpha}(0.5)$, $P^{(3)} = \boldsymbol{\alpha}(1)$, $P^{(4)} = \boldsymbol{\delta}(0.5)$, $P^{(5)} = \boldsymbol{\delta}(1)$, $P^{(6)} = \boldsymbol{\beta}(0.5)$, $P^{(7)} = \boldsymbol{\beta}(0)$, and $P^{(8)} = \boldsymbol{\gamma}(0.5)$. If a PDE patch $\mathbf{S}_h(u,v)$ shown in Fig. 1(b) is generated from the vertex-frame, its four corner vertices and four middle vertices

are $V^{(1)} = \mathbf{S}_h(0,0)$, $V^{(2)} = \mathbf{S}_h(0.5,0)$, $V^{(3)} = \mathbf{S}_h(1,0)$, $V^{(4)} = \mathbf{S}_h(1,0.5)$, $V^{(5)} = \mathbf{S}_h(1,1)$, $V^{(6)} = \mathbf{S}_h(0.5,1)$, $V^{(7)} = \mathbf{S}_h(0,1)$, and $V^{(8)} = \mathbf{S}_h(0,0.5)$. Since the eight curve vertices and the eight surface vertices indicate the same vertices, we have $V^{(k)} = P^{(k)}$ $(k = 1, 2, \ldots, 8)$.
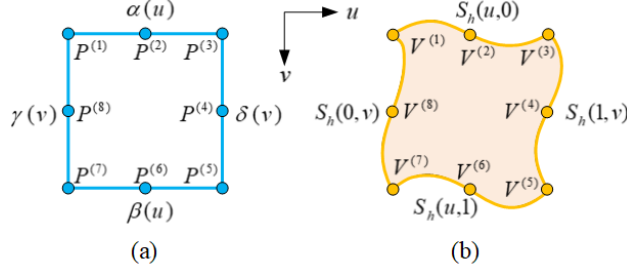


**Fig. 1** The vertex-frame (a) and the generated PDE patch (b).

For the sake of conciseness, the subscript "$h$" in the patch function $\mathbf{S}_h(u,v)$ is omitted in the remaining part of this subsection and the next subsection when more than one subscript is involved. That is to say, we use $S_{i,j}$ to stand for $S_{hi,j}$.

In order to use 4-sided PDE patches to create complicated 3D models, three different groups of 4-sided PDE patches are identified in Fig. 2. The first group contains only a single patch $S_{i,j}$, which is created first. The second group has four patches $S_{i-1,j}$, $S_{i,j+1}$, $S_{i+1,j}$, and $S_{i,j-1}$. Each of the four patches shares an edge highlighted in red with the patch $S_{i,j}$. The third group also has four patches $S_{i-1,j-1}$, $S_{i-1,j+1}$, $S_{i+1,j+1}$, and $S_{i+1,j-1}$. Each of them shares two edges highlighted in blue with its two adjacent patches. Generating these patches must follow the specified rule and direction. In the following subsection, we first introduce the patch generation rule, and then discuss patch direction.

### 3.2.1 Patch generation rule

Since the patch function $\mathbf{S}_h(u,v)$ is not the accurate solution of the PDE (3), substituting $\mathbf{S}_h(u,v)$ into PDE (3) will cause an error function $\mathbf{E}_h(u,v)$. If we uniformly allocate $M \times N$ points in the region $\{0 \le u \le 1; 0 \le v \le 1\}$, the $M \times N$ points can be represented by $(u_m, v_n)$ $\{1 \le m \le M; 1 \le n \le N\}$ where $u_m = \frac{m-1}{M-1}$ and $v_n = \frac{n-1}{N-1}$. The error of the PDE (3) at the point $(u_m, v_n)$ is denoted by $\mathbf{E}_h(u_m, v_n)$. We sum the squared errors at all these points, and obtain

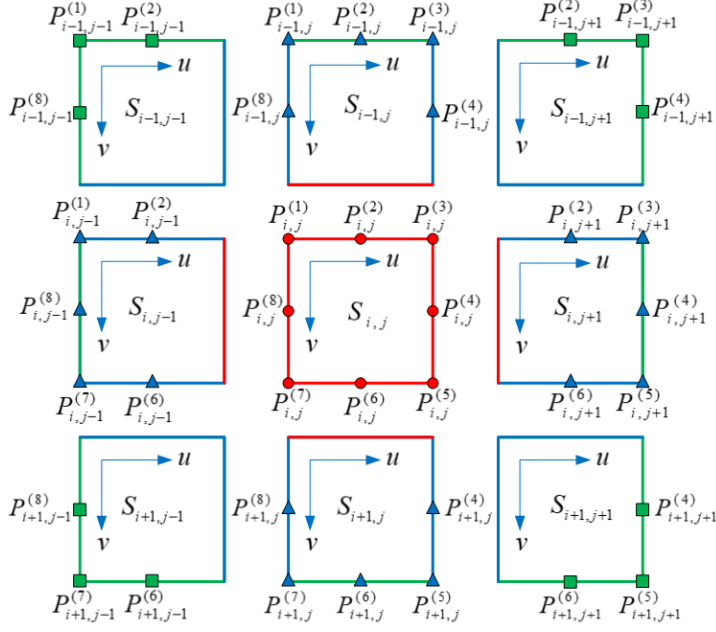$$J = \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{E}_h(u_m, v_n)^2 \tag{5}$$

**Fig. 2** PDE surface patches with vertex-frames

where

$$\mathbf{E}_h(u_m, v_n) = \mathbf{a_1}\frac{\partial^2 \mathbf{S}_h(u,v)}{\partial u^2} + \mathbf{a_2}\frac{\partial^2 \mathbf{S}_h(u,v)}{\partial v^2}\bigg|_{(u,v)=(u_m,v_n)} \tag{6}$$

Using the least squares method to minimize the squared error sum $J$ with respect to the two unknown constants $C^{(3)}$ and $C^{(6)}$ leads to the following equations

$$\frac{\partial J}{\partial C^{(3)}} = \sum_{m=1}^{M}\sum_{n=1}^{N} \mathbf{E}_h(u_m, v_n)\frac{\partial \mathbf{E}_h(u_m, v_n)}{\partial C^{(3)}} = 0$$

$$\frac{\partial J}{\partial C^{(6)}} = \sum_{m=1}^{M}\sum_{n=1}^{N} \mathbf{E}_h(u_m, v_n)\frac{\partial \mathbf{E}_h(u_m, v_n)}{\partial C^{(6)}} = 0 \tag{7}$$

In what follows, we use the conditions of the eight boundary vertices $V^{(k)} = P^{(k)}$ $(k = 1, 2, \ldots, 8)$ and Eq. (7) to determine all the unknown constants involved in the patch function (4) for each of the three groups of 4-sided patches shown in Fig. 2.

*First group of PDE patches* All the unknown constants in the PDE patch function (4) can be determined from the conditions of the eight boundary vertices $V_{i,j}^{(k)} = P_{i,j}^{(k)}$ $(k = 1, 2, \ldots, 8)$ and the two error minimization conditions

$\frac{\partial J_{i,j}}{\partial C_{i,j}^{(3)}} = 0$ and $\frac{\partial J_{i,j}}{\partial C_{i,j}^{(6)}} = 0$. By making Eq. (4) for Patch $S_{i,j}$ satisfy these ten conditions, we obtain all the ten unknown constants $C_{i,j}^{(k)} = \bar{P}_{i,j}^{(k)}$ ($k = 0, 1, 2, 3, ..., 9$) where $\bar{P}_{i,j}^{(k)}$ are given in Eq. (A.1).

*Second group of PDE patches* Each of this group of patches shares an edge (the red line in Fig. 2) with Patch $S_{i,j}$ whose unknown constants have been determined above. For Patch $S_{i,j+1}$, we have $S_{i,j+1}(0, v){=}S_{i,j}(1, v)$, i. e., $\boldsymbol{\gamma}_{i,j+1}(v) = \boldsymbol{\delta}_{i,j}(v)$, which is used to determine the three unknown constants $C_{i,j+1}^{(4)}, C_{i,j+1}^{(5)}$, and $C_{i,j+1}^{(6)}$. The remaining five boundary vertices $V_{i,j+1}^{(k)} = P_{i,j+1}^{(k)}$ ($k = 2, 3, ..., 6$) and the error minimization condition $\frac{\partial J_{i,j+1}}{\partial C_{i,j+1}^{(3)}} = 0$ are used to determine the remaining six unknown constants $C_{i,j+1}^{(k)}$ ($k = 1, 2, 3, 7, 8, 9$). All obtained unknown constants $C_{i,j+1}^{(k)} = \bar{P}_{i,j+1}^{(k)}$ ($k = 0, 1, 2, ..., 9$) are given in Eq. (A.4). The similar treatment is used to obtain the unknown constants of the remaining three 4-sided patches. For Patch $S_{i+1,j}$, the constraint conditions are $S_{i+1,j}(u, 0){=}S_{i,j}(u, 1)$, i. e., $\boldsymbol{\alpha}_{i+1,j}(u) = \boldsymbol{\beta}_{i,j}(u)$, $V_{i+1,j}^{(k)} = P_{i+1,j}^{(k)}$ ($k = 4, 5, ..., 8$) and the error minimization condition $\frac{\partial J_{i+1,j}}{\partial C_{i+1,j}^{(6)}} = 0$, which are used to determine the ten unknown constants $C_{i+1,j}^{(k)} = \bar{P}_{i+1,j}^{(k)}$ ($k = 0, 1, 2, ..., 9$) given in Eq. (A.6). For Patch $S_{i,j-1}$, the constraint conditions are $S_{i,j-1}(1, v){=}S_{i,j}(0, v)$, i. e., $\boldsymbol{\delta}_{i,j-1}(v) = \boldsymbol{\gamma}_{i,j}(v)$, $V_{i,j-1}^{(k)} = P_{i,j-1}^{(k)}$ ($k = 1, 2, 6, 7, 8$) and the error minimization condition $\frac{\partial J_{i,j-1}}{\partial C_{i,j-1}^{(3)}} = 0$, which are used to determine the ten unknown constants $C_{i,j-1}^{(k)} = \bar{P}_{i,j-1}^{(k)}$ ($k = 0, 1, 2, ..., 9$) given in Eq. (A.8). For Patch $S_{i-1,j}$, the constraint conditions are $S_{i-1,j}(u, 1){=}S_{i,j}(u, 0)$, i. e., $\boldsymbol{\beta}_{i-1,j}(u) = \boldsymbol{\alpha}_{i,j}(u)$, $V_{i-1,j}^{(k)} = P_{i-1,j}^{(k)}$ ($k = 1, 2, 3, 4, 8$) and the error minimization condition $\frac{\partial J_{i-1,j}}{\partial C_{i-1,j}^{(6)}} = 0$, which are used to determine the ten unknown constants $C_{i-1,j}^{(k)} = \bar{P}_{i-1,j}^{(k)}$ ($k = 0, 1, 2, ..., 9$) given in Eq. (A.10).

*Third group of PDE patches* Each of this group of patches share two edges highlighted in blue in Fig. 2 with two of the second group of patches whose unknown constants have been determined above. For Patch $S_{i+1,j+1}$, since it shares one edge highlighted in blue with Patch $S_{i,j+1}$ and another edge also highlighted in blue with Patch $S_{i+1,j}$ as shown in Fig. 2, we have $S_{i+1,j+1}(u, 0){=}S_{i,j+1}(u, 1)$, i. e., $\boldsymbol{\alpha}_{i+1,j+1}(u) = \boldsymbol{\beta}_{i,j+1}(u)$ and $S_{i+1,j+1}(0, v){=}S_{i+1,j}(1, v)$, i. e., $\boldsymbol{\gamma}_{i+1,j+1}(v) = \boldsymbol{\delta}_{i+1,j}(v)$, which are used to determine the seven unknown constants $C_{i+1,j+1}^{(k)}$ ($k = 0, 1, 2, ..., 6$). The remaining boundary vertices $V_{i+1,j+1}^{(k)} = P_{i+1,j+1}^{(k)}$ ($k = 4, 5, 6$) are used to determine the remaining three unknown constants $C_{i+1,j+1}^{(k)}$ ($k = 7, 8, 9$). By substituting Eq. (4) into these ten constraint conditions, we obtain the ten unknown constants $C_{i+1,j+1}^{(k)} = \bar{P}_{i+1,j+1}^{(k)}$ ($k = 0, 1, 2, ..., 9$), which are given in Eq. (A.12). For Patch $S_{i-1,j-1}$, since it shares one edge with Patch $S_{i-1,j}$ and another edge with $S_{i,j-1}$, we have $S_{i-1,j-1}(1, v){=}S_{i-1,j}(0, v)$, i. e.,

$\boldsymbol{\delta}_{i-1,j-1}(v) = \boldsymbol{\gamma}_{i-1,j}(v)$ and $S_{i-1,j-1}(u,1) = S_{i,j-1}(u,0)$, i. e., $\boldsymbol{\beta}_{i-1,j-1}(u) = \boldsymbol{\alpha}_{i,j-1}(u)$, and the boundary vertices $V_{i-1,j-1}^{(k)} = P_{i-1,j-1}^{(k)}$ $(k = 1, 2, 8)$. These ten constraint conditions are used to obtain the ten unknown constants $C_{i-1,j-1}^{(k)} = \bar{P}_{i-1,j-1}^{(k)}$ $(k = 0, 1, 2, ..., 9)$, which are given in Eq. (A.13). For Patch $S_{i-1,j+1}$, since it shares one edge with Patch $S_{i,j+1}$ and another edge with $S_{i-1,j}$, we have $S_{i-1,j+1}(u,1) = S_{i,j+1}(u,0)$, i. e., $\boldsymbol{\beta}_{i-1,j+1}(u) = \boldsymbol{\alpha}_{i,j+1}(u)$ and $S_{i-1,j+1}(0,v) = S_{i-1,j}(1,v)$, i. e., $\boldsymbol{\gamma}_{i-1,j+1}(v) = \boldsymbol{\delta}_{i-1,j}(v)$, and the boundary vertices $V_{i-1,j+1}^{(k)} = P_{i-1,j+1}^{(k)}$ $(k = 2, 3, 4)$. These ten constraint conditions are used to obtain the ten unknown constants $C_{i-1,j+1}^{(k)} = \bar{P}_{i-1,j+1}^{(k)}$ $(k = 0, 1, 2, ..., 9)$, which are given in Eq. (A.14). For Patch $S_{i+1,j-1}$, since it shares one edge with Patch $S_{i,j-1}$ and another edge with $S_{i+1,j}$, we have $S_{i+1,j-1}(u,0) = S_{i,j-1}(u,1)$, i. e., $\boldsymbol{\alpha}_{i+1,j-1}(u) = \boldsymbol{\beta}_{i,j-1}(u)$ and $S_{i+1,j-1}(1,v) = S_{i+1,j}(0,v)$, i. e., $\boldsymbol{\delta}_{i+1,j-1}(v) = \boldsymbol{\gamma}_{i+1,j}(v)$, and the boundary vertices $V_{i+1,j-1}^{(k)} = P_{i+1,j-1}^{(k)}$ $(k = 6, 7, 8)$. These ten constraint conditions are used to obtain the ten unknown constants $C_{i+1,j-1}^{(k)} = \bar{P}_{i+1,j-1}^{(k)}$ $(k = 0, 1, 2, ..., 9)$, which are given in Eq. (A.15).

### 3.2.2 Patch direction

The $uv$-coordinate systems for the PDE patches in different groups shown in Fig. 2 have the same $u$ and $v$ directions. Such a direction uniformity is suitable for creating regular and smooth 3D models. However, for a 3D model with a branching structure, the $u$ and $v$ directions of the $uv$-coordinate system for the branching structure are usually different from the $u$ and $v$ directions of the $uv$-coordinate system for the main body of the 3D model. For example, for the basket shown in Fig. 3 below, the $u$ and $v$ directions of the $uv$-coordinate system for the handle are opposite to the $u$ and $v$ directions of the $uv$-coordinate system for the main body. Such direction inconformity may lead to the following two situations in the direction parallel to the edge shared by two adjacent PDE patches: 1) the parametric variable of one patch is different from the parametric variable of the adjacent patch, 2) two adjacent patches share a same parametric variable but the direction of the parametric variable for one patch is opposite to the direction of the parametric variable for the adjacent patch. If we still use the same $u$ and $v$ directions shown in Fig. 2 to generate PDE patches for these two situations, position discontinuities on the edge occur. In order to avoid position discontinuities, it is necessary to adjust one parametric variable or the direction of one parametric variable for one of two adjacent patches in the direction parallel to the shared edge so that two adjacent PDE patches share a same parametric variable and a same direction in the direction parallel to their shared edge.

In total, the inconformity of parametric variables and directions of two adjacent patches in the direction parallel to the shared edge has six cases, which are shown in Fig. 4(a)-(f). In the figure, the PDE patches with odd number ($S_1$, $S_3$, $S_5$, $S_7$, $S_9$ and $S_{11}$) are known and the PDE patches with even number ($S_2$, $S_4$, $S_6$, $S_8$, $S_{10}$ and $S_{12}$) are unknown, which need to be generated. The six

**Fig. 3** An example of different patch directions (marked by dashed lines) of the basket model.

cases can be divided into three groups. The first group is shown in Fig. 4(a) and 4(d). In this group, both the parametric variables in the direction parallel to the shared edge and their directions for the two adjacent patches are different. Since the position continuity on the shared edge requires $S_2(u,0) = S_1(1,v)$ for the patches 1 and 2 with $v = 0$ corresponding to $u = 1$ and $S_8(0,v) = S_7(u,1)$ for the patches 7 and 8 with $u = 0$ corresponding to $v = 1$, we change the parametric variable for the top patches in the direction parallel to the shared edge to the other one and replace the changed parametric variable with 1 minus the changed parametric variable in the position continuity condition, i. e., $S_2(u,0) = S_1(1,1-u)$ and $S_8(0,v) = S_7(1-v,1)$. The second group is shown in Fig. 4(b) and 4(e). In this group, the parametric variables in the direction parallel to the shared edge are different but their directions for the two adjacent patches are the same. Since the position continuity on the shared edge requires $S_4(u,0) = S_3(0,v)$ for the patches 3 and 4 and $S_{10}(0,v) = S_9(u,0)$ for the patches 9 and 10, we change the parametric variable for the top patches in the direction parallel to the shared edge to the other one in the position continuity condition, i. e., $S_4(u,0) = S_3(0,u)$ and $S_{10}(0,v) = S_9(v,0)$. The third group is shown in Fig. 4(c) and 4(f). In this group, the parametric variables in the direction parallel to the shared edge are the same but their directions for the two adjacent patches are opposite. Since the position continuity on the shared edge requires $S_6(u,0) = S_5(u,0)$ for the patches 5 and 6 and $S_{12}(0,v) = S_{11}(0,v)$ for the patches 11 and 12, we replace the parametric variable with 1 minus the parametric variable for the top patches in the position continuity condition, i. e., $S_6(u,0) = S_5(1-u,0)$, and $S_{12}(0,v) = S_{11}(0,1-v)$.

### 3.3 Particular solution of second-order nonhomogeneous PDE

As discussed previously, the right-hand side term of PDE (1) act as a sculpting force to easily and effectively create different shapes of a PDE surface. The solution corresponding to the right-hand side term of PDE (1) is a particular solution. In this subsection, we investigate the particular solution of PDE (1).

When deriving the general solution $\mathbf{S}_h(u,v)$ of the vector-valued second-order homogeneous PDE (3), the positional continuous boundary conditions
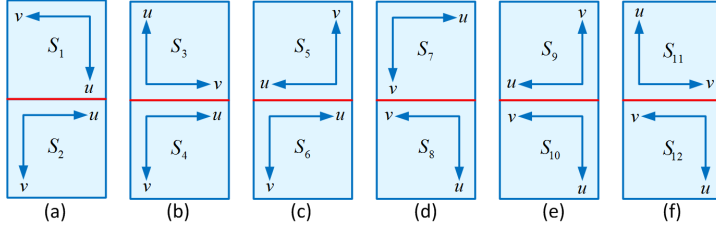
**Fig. 4** Six forms of direction inconformity of two adjacent patches

(2) have been exactly satisfied. Therefore, when deriving the particular solution $\mathbf{S}_p(u, v)$ of the vector-valued second-order nonhomogeneous PDE (1), the boundary conditions (2) should not be changed. It means the particular solution must satisfy the following zeroed boundary conditions

$$\mathbf{S}_p(u, 0) = 0, \mathbf{S}_p(u, 1) = 0 \ \forall u \in [0, 1]$$
$$\mathbf{S}_p(0, v) = 0, \mathbf{S}_p(1, v) = 0 \ \forall v \in [0, 1] \tag{8}$$

The vector-valued sculpting force function can take different forms. Since the particular solution depending on the vector-valued sculpting force function must satisfy the requirements of both PDE (1) and boundary conditions (8), the construction of the vector-valued sculpting force function must take these requirements into account. There are different functions that can satisfy these requirements. Here we take the following function as the vector-valued sculpting force function

$$\mathbf{F}(u, v) = \mathbf{f}_0 sin(\pi u) sin(\pi v) \tag{9}$$

where $\mathbf{f}_0 = [f_{0x} \ f_{0y} \ f_{0z}]^T$ is a vector-valued coefficient whose direction and size can be changed to apply different sculpting forces to a PDE surface.

According to the vector-valued sculpting force function (9), the vector-valued particular solution can be taken to be

$$\mathbf{S}_p(u, v) = \mathbf{s}_{p0} sin(\pi u) sin(\pi v) \tag{10}$$

Clearly, the above particular solution (10) exactly satisfies the zeroed boundary conditions (8). The remaining problem is to satisfy the PDE (1). Using $\mathbf{S}_p(u, v)$ in (10) to replace $\mathbf{S}(u, v)$ in (1) and introducing (9) into (1), we obtain

$$\mathbf{s}_{p0} = -\frac{\mathbf{f}_0}{\pi^2(\mathbf{a}_1 + \mathbf{a}_2)} \tag{11}$$

Substituting Eq. (11) into Eq. (10), the particular solution of the vector-valued second-order nonhomogeneous PDE (1) is obtained as

$$\mathbf{S}_p(u, v) = -\frac{\mathbf{f}_0}{\pi^2(\mathbf{a}_1 + \mathbf{a}_2)} sin(\pi u) sin(\pi v) \tag{12}$$

Putting the obtained general solution of the vector-valued second-order homogeneous PDE (3) and the particular solution (12) of the vector-valued

second-order nonhomogeneous PDE (1) together, we obtain the following general solution of the vector-valued second-order nonhomogeneous PDE (1)

$$
\begin{aligned}
\mathbf{S}(u,v) = \mathbf{S}_h(u,v) + \mathbf{S}_p(u,v) = {} & \bar{P}_{\bar{m},\bar{n}}^{(0)} + \bar{P}_{\bar{m},\bar{n}}^{(1)}u + \bar{P}_{\bar{m},\bar{n}}^{(2)}u^2 \\
& + \bar{P}_{\bar{m},\bar{n}}^{(3)}u^3 + \bar{P}_{\bar{m},\bar{n}}^{(4)}v + \bar{P}_{\bar{m},\bar{n}}^{(5)}v^2 + \bar{P}_{\bar{m},\bar{n}}^{(6)}v^3 + \bar{P}_{\bar{m},\bar{n}}^{(7)}uv \\
& + \bar{P}_{\bar{m},\bar{n}}^{(8)}u^2v + \bar{P}_{\bar{m},\bar{n}}^{(9)}uv^2 - \frac{\mathbf{f}_0}{\pi^2(\mathbf{a}_1 + \mathbf{a}_2)}sin(\pi u)sin(\pi v)
\end{aligned}
\tag{13}
$$

where $\bar{P}_{\bar{m},\bar{n}}^{(k)}$ $(k = 0, 1, 2, ..., 9; \bar{m} = i - 1, i, i + 1; \bar{n} = j - 1, j, j + 1)$ are the analytical formulae obtained in Subsection 3.2.1 and given in A.

The above general solution defines an analytical $C^0$ continuous 4-sided PDE patch. It can be used to create various shapes of a single patch and generate complicated 3D models to be demonstrated in the following two sections.

## 4 Results and comparisons

We validated our approach using a set of test models that includes both man-made (the dress, mug and hat) and organic (the human face and ear) models as shown in Fig. 5. These vertex-frame models are generated by extracting feature curves from free 3D objects in TURBOSQUID (http://www.turbosquid.com). The shape of a PDE patch-based surface model generated from a vertex-frame model is controlled by three kinds of parameters: the position coordinate of eight vertices $P^{(k)}$ $(k = 1, 2, ..., 8)$ used to define the boundary of a PDE patch, the two shape control parameters $\mathbf{a}_1$ and $\mathbf{a}_2$ and sculpting force used to control surface shapes.
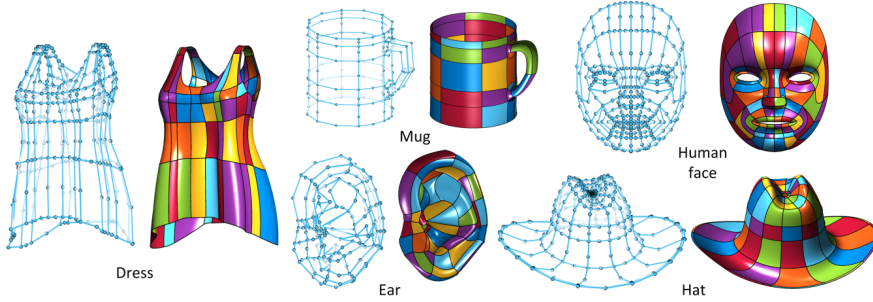


**Fig. 5** Test results of examples. Each example includes its vertex-frame model (left) and PDE patch-base surface model (right)

### 4.1 Comparison of accuracy and data size

We compare our proposed method with polygon and NURBS modeling techniques. Since the 3D objects used in our tests come from real models, these
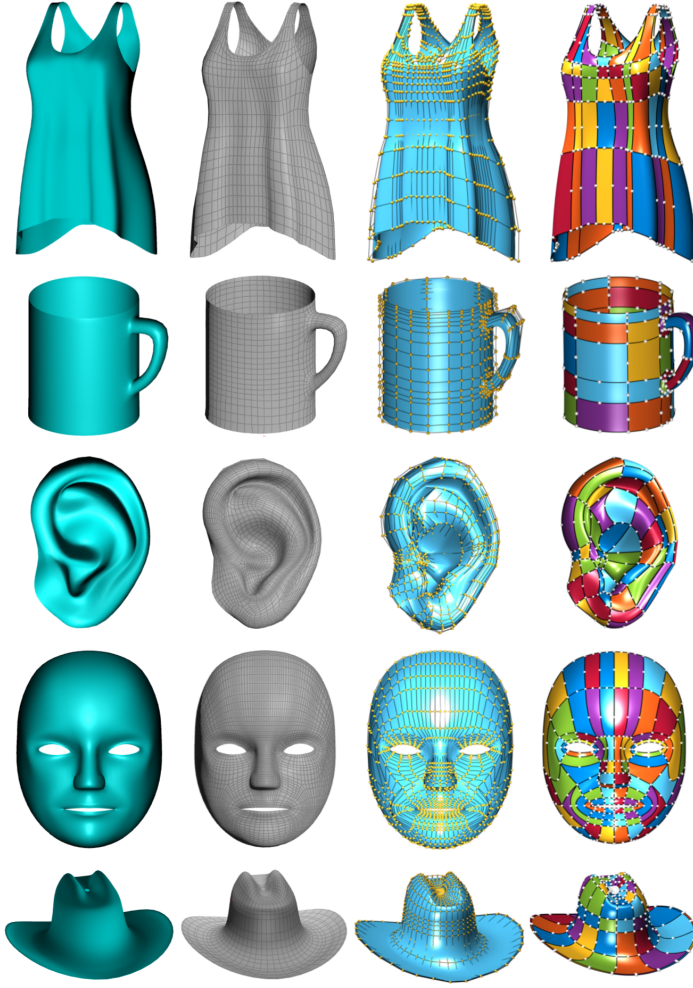
**Fig. 6** Comparisons with polygon and NURBS modeling

real models can be regarded as the ground truth in our comparisons. Fig. 6 shows the five ground truth models (the first column) rebuilt by polygon (the second column), NURBS (the third column) and our proposed 4-sided PDE patches (the fourth column), respectively. In order to compare the storage requirements of the three modeling techniques, the corresponding three rebuilt models should have the same or very close accuracy against the ground truth model. We use the root mean square (RMS) error with respect to the bounding box diagonal, which is evaluated with Metro tool [44] to measure the accuracy of the created models. The RMS error is evaluated between the sample points of the rebuilt models and the ground truth model. Our experiment indicates that the number of sample points same as the vertex number of the rebuilt

polygon model gives very accurate results. The comparison needs to satisfy the following criteria. 1) RMS errors between the three models and the ground truth model are taken to be the same or very close. 2) The data of the polygon model are the coordinate values of vertices, the data of the NURBS model are the coordinate values of control points and weights, and the data of our PDE model are coordinate values of boundary vertices, the two shape control parameters, and the vector-valued coefficient of the sculpting force defined in Eq. (9). 3) The values of all data used in the comparison are rounded to six significant digits.

Table 1 lists the statistical data of the three modeling techniques for the five surface models, including the number of vertices or control points, RMS errors and storage amounts. According to Table 1, for the surface model of the dress, mug, ear, hat and human face, our proposed modeling of 4-sided PDE patches saves the storage space of 81.3%, 79.5%, 85.2%, 78.3% and 85.9% over the polygon modeling, and 58.0%, 56.6%, 57.6%, 58.1%, and 58.4% over the NURBS modeling. Therefore, our proposed method has a much smaller requirement for storage space. Moreover, the results indicate our proposed method only need a small amount of vertices to generate surface models with almost same RMS errors compared to polygon and NURBS techniques. It means our proposed method is more efficient and easier for users to do manual operations and real-time interactive design. This will be discussed in detail in Section 5.

**Table 1** Statistical data of polygon (PO), NURBS (NU) and our proposed method for the five surface models

|       | Vertices or control points | | | RMS error (%) | | | Storage (KB) | | |
|-------|------|------|------|------|-----|------|------|------|------|
|       | PO   | NU   | Ours | PO   | NU  | Ours | PO   | NU   | Ours |
| Dress | 2996 | 1020 | 362  | 0.1  | 0.1 | 0.1  | 76.0 | 33.8 | 14.2 |
| Mug   | 1275 | 460  | 168  | 0.2  | 0.2 | 0.2  | 32.2 | 15.2 | 6.6  |
| Ear   | 1862 | 498  | 174  | 0.4  | 0.4 | 0.4  | 47.2 | 16.5 | 7.0  |
| Hat   | 2256 | 900  | 312  | 0.1  | 0.1 | 0.1  | 58.5 | 30.3 | 12.7 |
| Face  | 3701 | 963  | 336  | 0.1  | 0.1 | 0.1  | 94.8 | 32.2 | 13.4 |

4.2 Comparison of the local deformation

Since the sculpting force of PDE-based modeling is powerful in creating different deformations, we compare shape manipulation of our proposed method with NURBS and Coons modeling techniques to demonstrate the capacity and effectiveness of our proposed method.

Fig. 7 shows the comparison of the interior deformation between PDE and NURBS patches. We generate a simple PDE patch (a) and a NURBS patch (c) with a same surface shape. The PDE patch can easily produce a small convex shape (b) within arbitrary inner local regions by interactively manipulating a sculpting force, but the NURBS patch with the given control points cannot

create such a small convex shape through interactively moving its control points, as shown in (d). In order to obtain the similar shape to that shown in (b), the most common approach is to interpolate the NURBS patch in the deformation region by adding new control points, as shown in (e). However, the interpolation needs extra manual operations and will greatly increase the data size of the NURBS patch.
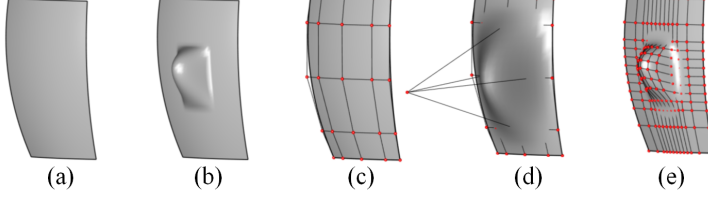


**Fig. 7** Comparison of the interior deformation between PDE and NURBS patches. (a) An original PDE patch. (b) The PDE patch (a) with a local deformation created by manipulating a sculpting force. (c) An original NURBS patch with the same shape as (a). (d) The deformation of the NURBS patch (c) by moving control points. (e) The deformation of the NURBS patch (c) after the interpolation.

PDE-based and Coons modeling techniques are both boundary-based surface creation methods. Since bilinear Coons surfaces have no shape manipulation capacity, and bicubic Coons surfaces can manipulate surface shapes by changing boundary tangents, we compare our proposed method with bicubic Coons surfaces in manipulating surface shapes. The sculpting force in our proposed method can deform: 1) the whole PDE patch (Fig. 8(b)), 2) inner local regions of a single PDE patch (8(c) and (d)) by controlling the range of parametric variables $u$ and $v$ in $\mathbf{S}_p(u, v)$, and 3) arbitrary regions across multiple PDE patches (Fig. 9). In contrast, boundary tangents of a bicubic Coons patch can only deform the whole region rather than arbitrary inner local regions of the patch. Fig. 8 shows the comparison between PDE patches ((a)-(d)) and bicubic Coons patches ((e)-(h)) with the four same boundaries $\boldsymbol{\alpha}(u)$, $\boldsymbol{\beta}(u)$, $\boldsymbol{\gamma}(v)$ and $\boldsymbol{\delta}(v)$. We apply $w_1\boldsymbol{\alpha}(u)$, $w_2\boldsymbol{\beta}(u)$, $w_3\boldsymbol{\gamma}(v)$ and $w_4\boldsymbol{\delta}(v)$ to describe the boundary tangents of bicubic Coons patches where $w_i = [w_{ix}, w_{iy}, w_{iz}]^T$ $(i = 1, 2, 3, 4)$. The results show that our proposed method is more flexible and powerful in shape manipulations and local deformations.

## 5 Interactive design

5.1 Shape control parameter and sculpting force

The shape of the PDE patch generated from a given vertex-frame can be effectively controlled by two shape control parameter $\mathbf{a}_1$, $\mathbf{a}_2$ and a sculpting force $\mathbf{f}_0$. We develop a tool shown in Fig. 10 for interactively choosing their suitable values. After inputting a vertex-frame, different shapes of a PDE patch
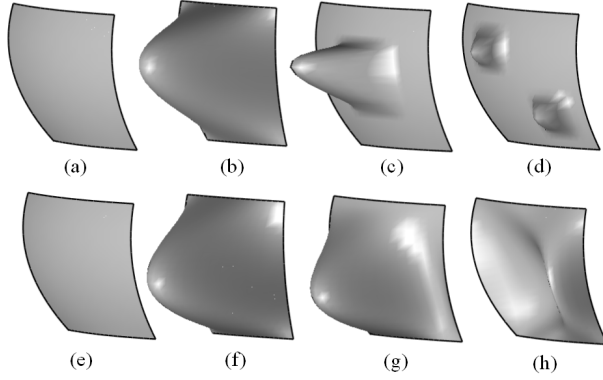
**Fig. 8** The comparison between PDE surface patches (a)-(d) and bicubic Coons surface patches (e)-(h). (a) $\mathbf{f}_0 = [0\ 0\ 0]$. (b) $\mathbf{f}_0 = [20\ 1\ 1]$; (c) $\mathbf{f}_0 = [3\ 1\ 2]$ and the deformation region $0.25 \leq u \leq 0.75$, $0.25 \leq v \leq 0.75$. (d) $\mathbf{f}_0 = [2\ 0\ 0]$ and deformation region $0.2 \leq u \leq 0.45$, $0.2 \leq v \leq 0.45$ and $0.6 \leq u \leq 0.9$, $0.55 \leq v \leq 0.8$. (e) $w_1 = w_2 = w_3 = w_4 = [0\ 0\ 0]$. (f) $w_1 = w_3 = [100\ 0\ 0]$, $w_2 = w_4 = [-100\ 0\ 0]$. (g) $w_1 = w_2 = w_3 = [0\ 0\ 0]$, $w_4 = [-100\ 0\ 0]$. (h) $w_1 = [-20\ 0\ -20]$, $w_2 = [-20\ 0\ 0]$, $w_3 = w_4 = [0\ 0\ 0]$.
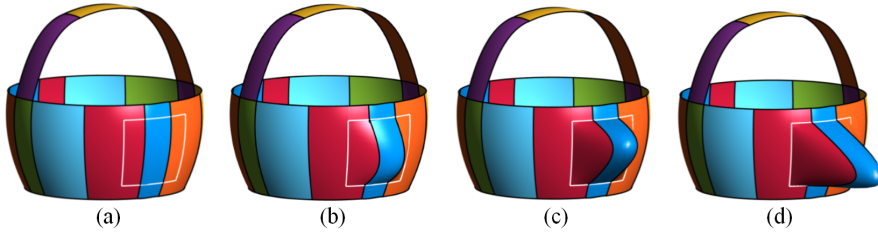


**Fig. 9** The shape deformation in a region encircled by boundary curves (white) across multiple PDE patches. (a) $\mathbf{f}_0 = [0\ 0\ 0]$. (b) $\mathbf{f}_0 = [-1\ -1\ 1]$. (c) $\mathbf{f}_0 = [-5\ 1\ 0]$. (d) $\mathbf{f}_0 = [-8\ -2\ 2]$.

can be generated by moving the slider of different components of $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{f}_0$. The default range of each slider is $[0, 1]$, and there is an input box next to each slider for inputting a magnification coefficient. The values of $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{f}_0$ are displayed in real time under the sliders.

The function of shape control parameters $\mathbf{a}_1$ and $\mathbf{a}_2$ is to weight the first and second terms on the left-hand side of the PDE (1) and create a smoothing effect to the differential operator in $u$ and $v$ direcions [28]. The two parameters can be used to directly change the shape of PDE patches. As shown in Fig. 11, we change the shape control parameters $\mathbf{a}_1^{left}$ and $\mathbf{a}_2^{left}$ of the patch at the left joint of the basket strap and $\mathbf{a}_1^{right}$ and $\mathbf{a}_2^{right}$ of the patch at the right joint to demonstrate the effect of the two shape control parameters. It can be seen from the figure that by changing the two shape control parameters, we can
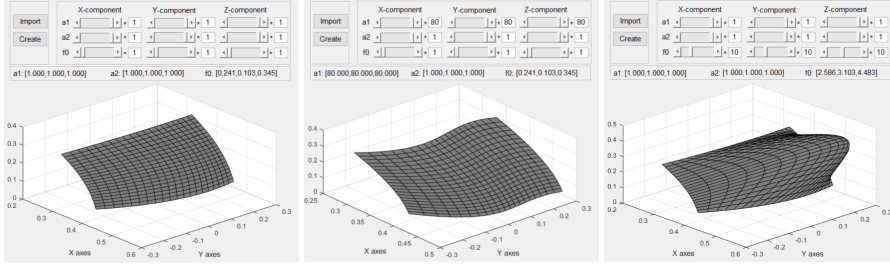
**Fig. 10** The tool for interactively manipulating $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{f}_0$.
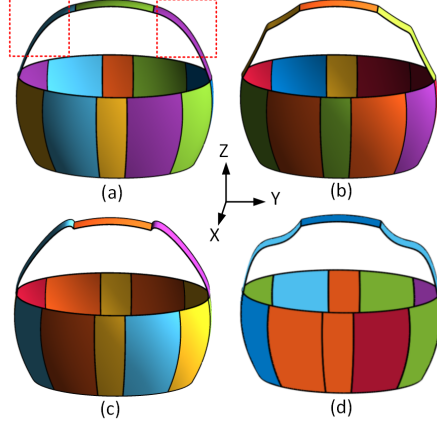


**Fig. 11** Design parameter modification of a basket. (a) $\mathbf{a}_1^{left} = \mathbf{a}_2^{left} = \mathbf{a}_1^{right} = \mathbf{a}_2^{right} = [1 \ 1 \ 1]^T$. (b) $\mathbf{a}_1^{left} = \mathbf{a}_1^{right} = 150 \times [1 \ 1 \ 1]^T$, $\mathbf{a}_2^{left} = \mathbf{a}_2^{right} = [1 \ 1 \ 1]^T$. (c) $\mathbf{a}_1^{left} = \mathbf{a}_1^{right} = -100 \times [1 \ 1 \ 1]^T$, $\mathbf{a}_2^{left} = \mathbf{a}_2^{right} = [1 \ 1 \ 1]^T$. (d) $\mathbf{a}_1^{left} = \mathbf{a}_1^{right} = [1 \ 150 \ 1]^T$, $\mathbf{a}_2^{left} = \mathbf{a}_2^{right} = [1 \ 1 \ 1]^T$.

obtain various shapes of the basket strap without moving the vertex positions of vertex-frames.

## 5.2 Interactive tool

Besides the shape control parameters and sculpting forces, which can be used in surface manipulations, the shape of a PDE surface patch can also be changed by the positions of eight vertices of its vertex-frame. When users are dissatisfied with some shapes of an output surface model, it is time-consuming to remake a new version of the corresponding vertex-frames and create new shapes to replace the unsatisfactory shapes. In contrast, modifying the corresponding vertices of vertex-frames to obtain their desired shapes is easier and will greatly improve the modeling efficiency. In order to expand the practicability of our proposed method, the user should be able to modify the vertex-frame model
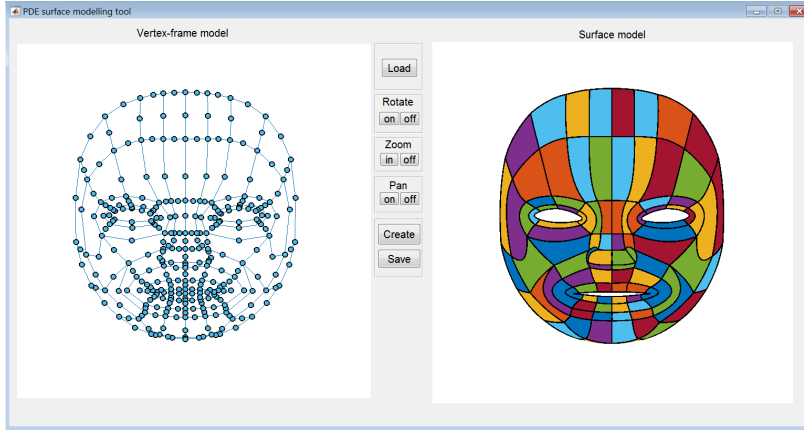
**Fig. 12** Interactive tool of PDE patch-based surface modeling.

and generate the desired surface model in real time. Therefore, we design an interactive interface with which users can move every vertex of an input vertex-frame model and check the quality of an output surface model. The interface of our interactive tool is shown in Fig. 12, which includes three transformation functions, i. e., rotation, moving and scaling, for both the vertex-frame model and surface model.

The interactive tool is developed in the Cartesian space. The user can manipulate the models by using the three transformation functions. After inputting a vertex-frame model by clicking the Load button, the vertex-frame model is presented in left position. The user can move a vertex by clicking and dragging it with mouse, and releasing the mouse button at desired position (Fig. 13). When the modification ends, the user can click the Create button to generate a surface model in real time and Save button to save the surface model.
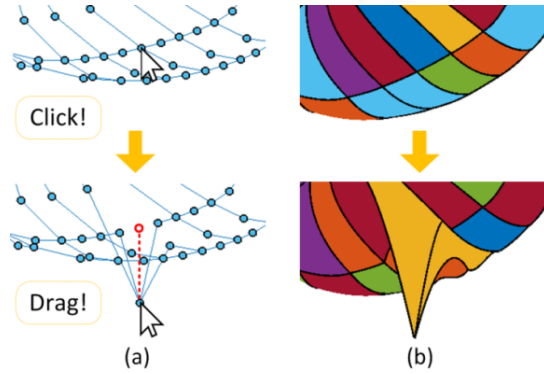


**Fig. 13** Editing of a vertex. (a) the vertex-frame model. (b) the surface model.

We invited three volunteers to design different surface models using our interactive tool. All of them have experience in 3D modeling tasks. The test objects are the five vertex-frame models shown in Fig. 5. We explained how to use the interactive tool in a ten-minute training session and then asked them to design and recreate new surface models based on the five existing vertex-frame models. Fig. 14 shows the results created by the three volunteers. They generated some interesting surface models by modifying only a few vertices of the corresponding vertex-frame models. The feedback from the volunteers was very positive. All of them believed that our proposed method is easier to create and manipulate desired surface models than polygon and NURBS modeling techniques.



**Fig. 14** Results created by the three volunteers.

## 6 Conclusions and future work

We have developed the first analytical $C^0$ continuous 4-sided PDE patches to tackle the weaknesses of big data and heavy manual operations of polygon and NURBS modeling, and integrate shape control parameters and sculpting forces in boundary-based PDE patches to avoid weak ability of Coons surfaces in shape manipulations. We have also developed a novel PDE patch-based surface modeling technique to quickly create 3D models from vertex-frames, and implemented the modeling technique into an interactive software tool to input and modify vertex-frame models and create and manipulate surface models in real time. The experiments made by testing several vertex-frame models validate the capacity and effectiveness of our proposed method and

demonstrate that our proposed method can create complex surface models with a small number of patches and fewer manual operations. Moreover, we have compared our proposed method with polygon and NURBS techniques in creating different surface models with respect to the ground-truth model. The results indicate that our proposed method greatly reduces the data size of 3D models and save storage space. We have also compared our proposed method with Coons surfaces in manipulating surface shapes and demonstrated that our proposed method is more flexible and powerful.

Our proposed method has some limitations. Firstly, 4-sided PDE surface patches are not flexible enough to represent complex shapes of some 3D models such as branching structures. For such structures, 4-sided patches combined with 3-sided patches are easier and more powerful to create their surface models and reduce the number of patches. Secondly, the method developed in this paper can only achieve position continuity between adjacent PDE patches. 3- and 4-sided PDE patches with tangent continuity, curvature continuity, and even higher order $C^n$ continuity have not been investigated in this paper. Thirdly, how to use sculpting forces to create local deformations in the regions bounded by complicated boundary curves have not been demonstrated. Fourthly, the patch generation rule and patch direction discussed in Subsections 3.2.1 and 3.2.2 must be followed when creating 3D models from the analytical $C^0$ continuous 4-sided PDE patches. In our following work, we will develop $C^n$ continuous 3- and 4- sided PDE patches without the constraints of the patch generation rule and patch direction and investigate how to use sculpting forces to create different deformations in the regions bounded by various complicated boundary curves to demonstrate the advantages of PDE-based geometric modeling over polygon and NURBS modeling in shape creation and manipulation.

## Conflict of interest

The authors declare that they have no conflict of interest.

## References

1. Mario Russo. *Polygonal modeling: basic and advanced techniques.* Jones & Bartlett Learning, 2006.
2. Les Piegl and Wayne Tiller. *The NURBS book.* Springer Science & Business Media, 2012.
3. Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978.

4. Daniel Doo and Malcolm Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978.
5. David Salomon. *Curves and surfaces for computer graphics.* Springer Science & Business Media, 2007.
6. Marcelo Epstein. Partial differential equations in engineering. In *Partial Differential Equations*, pages 25–47. Springer, 2017.
7. Jian J Zhang and Lihua You. Pde based surface representation-vase design. *Computers & Graphics*, 26(1):89–98, 2002.
8. Haixia Du and Hong Qin. Direct manipulation and interactive sculpting of pde surfaces. In *Computer Graphics Forum*, volume 19, pages 261–270. Wiley Online Library, 2000.
9. Haixia Du and Hong Qin. Dynamic pde-based surface design using geometric and physical constraints. *Graphical Models*, 67(1):43–71, 2005.
10. Joanna M Brown, Malcolm IG Bloor, M Susan Bloor, and Michael J Wilson. The accuracy of b-spline finite element approximations to pde surfaces. *Computer methods in applied mechanics and engineering*, 158(3-4):221–234, 1998.
11. Alec Jacobson, Elif Tosun, Olga Sorkine, and Denis Zorin. Mixed finite elements for variational surface modeling. In *Computer graphics forum*, volume 29, pages 1565–1574. Wiley Online Library, 2010.
12. Dan Liu and Guoliang Xu. A general sixth order geometric partial differential equation and its application in surface modeling. *Journal of Information and Computational Science*, 4(1):129–140, 2007.
13. Haixia Du and Hong Qin. A shape design system using volumetric implicit pdes. In *ACM SIGGRAPH 2005 Courses*, page 116. ACM, 2005.
14. Jisoon Park, Taewon Kim, Seung-Yeob Baek, and Kunwoo Lee. An algorithm for estimating surface normal from its boundary curves. *Journal of Computational Design and Engineering*, 2(1):67–72, 2015.
15. Tibor Stanko, Stefanie Hahmann, Georges-Pierre Bonneau, and Nathalie Saguin-Sprynski. Surfacing curve networks with normal control. *Computers & Graphics*, 60:1–8, 2016.
16. Qing Pan, Guoliang Xu, and Yongjie Zhang. A unified method for hybrid subdivision surface design using geometric partial differential equations. *Computer-Aided Design*, 46:110–119, 2014.
17. Robert Schneider and Leif Kobbelt. Geometric fairing of irregular meshes for free-form surface design. *Computer aided geometric design*, 18(4):359–379, 2001.
18. Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2008.
19. Haixia Du and Hong Qin. Free-form geometric modeling by integrating parametric and implicit pdes. *IEEE Transactions on Visualization & Computer Graphics*, (3):549–561, 2007.
20. Guoliang Xu and Qin Zhang. A general framework for surface modeling using geometric partial differential equations. *Computer Aided Geometric Design*, 25(3):181–202, 2008.

21. Malcolm IG Bloor and Michael J Wilson. Using partial differential equations to generate free-form surfaces. *Computer-Aided Design*, 22(4):202–212, 1990.

22. Jian J Zhang and LH You. Fast surface modelling using a 6th order pde. In *Computer Graphics Forum*, volume 23, pages 311–320, 2004.

23. Hassan Ugail, Malcolm IG Bloor, and Michael J Wilson. Techniques for interactive design using the pde method. *ACM Transactions on Graphics (TOG)*, 18(2):195–212, 1999.

24. Michael Athanasopoulos, Hassan Ugail, and Gabriela González Castro. Parametric design of aircraft geometry using partial differential equations. *Advances in Engineering Software*, 40(7):479–486, 2009.

25. MIG Bloor and MJ Wilson. Generating blend surfaces using partial differential equations. *Computer-Aided Design*, 21(3):165–171, 1989.

26. Lihua You, Jian J Zhang, and Peter Comninos. Blending surface generation using a fast and accurate analytical solution of a fourth-order pde with three shape control parameters. *The Visual Computer*, 20(2-3):199–214, 2004.

27. Lihua You and Jian J. Zhang. Fast generation of 3-d deformable moving surfaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(4):616–625, 2003.

28. Sebastian Kubiesa, Hassan Ugail, and Michael Wilson. Interactive design using higher order pdes. *The Visual Computer*, 20(10):682–693, 2004.

29. LH You, Peter Comninos, and Jian J Zhang. Pde blending surfaces with c2 continuity. *Computers & Graphics*, 28(6):895–906, 2004.

30. Malcolm IG Bloor and Michael J Wilson. An analytic pseudo-spectral method to generate a regular 4-sided pde surface patch. *Computer Aided Geometric Design*, 22(3):203–219, 2005.

31. Yun Sheng, Alexei Sourin, Gabriela Gonzalez Castro, and Hassan Ugail. A pde method for patchwise approximation of large polygon meshes. *The Visual Computer*, 26(6-8):975–984, 2010.

32. Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Siggraph*, volume 98, pages 395–404. Citeseer, 1998.

33. Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94. Citeseer, 1998.

34. Joe Warren and Henrik Weimer. *Subdivision methods for geometric design: A constructive approach*. Elsevier, 2001.

35. Ursula Augsdörfer. Nurbs with extraordinary points: high-degree non-uniform subdivision surfaces. *ACM transactions on graphics*, 28(3; 46):46–1, 2009.

36. Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

37. Tereza Flaxman. *Maya 2008 Character Modeling and animation: Principles and practices*. Cengage Learning, 2008.

38. Gerald Farin and Dianne Hansford. Discrete coons patches. *Computer Aided Geometric Design*, 16(7):691–700, 1999.
39. Steven A Coons. Surfaces for computer-aided design of space forms. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, 1967.
40. Malcolm IG Bloor, Michael J Wilson, and H Hagen. The smoothing properties of variational schemes for surface design. *Computer Aided Geometric Design*, 12(4):381–394, 1995.
41. Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. Citeseer, 1999.
42. MIG Bloor and MJ Wilson. Blend design as a boundary-value problem. In *Theory and practice of geometric modeling*, pages 221–234. Springer, 1989.
43. Malcolm IG Bloor and Michael J Wilson. Functionality in solids obtained from partial differential equations. In *Geometric modelling*, pages 21–42. Springer, 1993.
44. Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: measuring error on simplified surfaces. In *Computer Graphics Forum*, volume 17, pages 167–174. Wiley Online Library, 1998.

# A Analytical formulae of $\bar{P}_{\bar{m},\bar{n}}^{(k)}$ ($\bar{m} = i-1,\ i,\ i+1;\ \bar{n} = j-1,\ j,\ j+1$)

For $S_{i,j}$ patch, $\bar{P}_{i,j}^{(k)}$ ($k = 0, 1, 2, 3, ..., 9$) are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i,j}^{(0)} = & P_{i,j}^{(1)} \\
\bar{P}_{i,j}^{(1)} = & -3P_{i,j}^{(1)} + 4P_{i,j}^{(2)} - P_{i,j}^{(3)} + \frac{1}{2}\bar{P}_{i,j}^{(3)} \\
\bar{P}_{i,j}^{(2)} = & 2P_{i,j}^{(1)} - 4P_{i,j}^{(2)} + 2P_{i,j}^{(3)} - \frac{3}{2}\bar{P}_{i,j}^{(3)} \\
\bar{P}_{i,j}^{(3)} = & \frac{\bar{A}_{i,j}D_{i,j} - A_{i,j}\bar{B}_{i,j}}{\bar{B}_{i,j}B_{i,j} - D_{i,j}D_{i,j}} \\
\bar{P}_{i,j}^{(4)} = & 4P_{i,j}^{(8)} - P_{i,j}^{(7)} - 3P_{i,j}^{(1)} + \frac{1}{2}\bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j}^{(5)} = & -4P_{i,j}^{(8)} + 2P_{i,j}^{(7)} + 2P_{i,j}^{(1)} - \frac{3}{2}\bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j}^{(6)} = & \frac{A_{i,j}D_{i,j} - \bar{A}_{i,j}\bar{B}_{i,j}}{\bar{B}_{i,j}B_{i,j} - D_{i,j}D_{i,j}} \\
\bar{P}_{i,j}^{(7)} = & 5P_{i,j}^{(1)} - 4P_{i,j}^{(2)} - P_{i,j}^{(3)} + 4P_{i,j}^{(4)} - 3P_{i,j}^{(5)} + 4P_{i,j}^{(6)} \\
& - P_{i,j}^{(7)} - 4P_{i,j}^{(8)} \\
\bar{P}_{i,j}^{(8)} = & -2P_{i,j}^{(1)} + 4P_{i,j}^{(2)} - 2P_{i,j}^{(3)} + 2P_{i,j}^{(5)} - 4P_{i,j}^{(6)} + 2P_{i,j}^{(7)} \\
\bar{P}_{i,j}^{(9)} = & -2P_{i,j}^{(1)} + 2P_{i,j}^{(3)} - 4P_{i,j}^{(4)} + 2P_{i,j}^{(5)} - 2P_{i,j}^{(7)} + 4P_{i,j}^{(8)}
\end{cases}
\tag{A.1}
$$

where

$$B_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} a_1^2 (6u_m - 3)^2$$

$$\bar{B}_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} a_2^2 (6v_n - 3)^2$$

$$D_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} a_1 a_2 (6u_m - 3)(6v_n - 3) \qquad (A.2)$$

$$A_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} a_1 (6u_m - 3) F_{i,j}^{m,n}$$

$$\bar{A}_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} a_2 (6v_n - 3) F_{i,j}^{m,n}$$

and $F_{i,j}^{m,n}$ is determined by:

$$
\begin{aligned}
F_{i,j}^{m,n} =\,& 4(a_1 - a_1 v_n + a_2 - a_2 u_m) P_{i,j}^{(1)} \\
& + 8a_1(v_n - 1) P_{i,j}^{(2)} + 4(a_1 - a_1 v_n + a_2 u_m) P_{i,j}^{(3)} \\
& - 8a_2 u_m P_{i,j}^{(4)} + 4(a_1 v_n + a_2 u_m) P_{i,j}^{(5)} - 8a_1 v_n P_{i,j}^{(6)} \\
& + 4a_2(1 - u_m) P_{i,j}^{(7)} + 8a_2 u_m P_{i,j}^{(8)}
\end{aligned}
\qquad (A.3)
$$

For $S_{i,j+1}$ patch, $\bar{P}_{i,j+1}^{(k)}$ ($k = 0, 1, 2, 3, ..., 9$) are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i,j+1}^{(0)} = & \bar{P}_{i,j}^{(0)} + \bar{P}_{i,j}^{(1)} + \bar{P}_{i,j}^{(2)} + \bar{P}_{i,j}^{(3)} \\
\bar{P}_{i,j+1}^{(1)} = & 4P_{i,j+1}^{(2)} - P_{i,j+1}^{(3)} - 3\bar{P}_{i,j+1}^{(0)} + \frac{1}{2}\bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i,j+1}^{(2)} = & -4P_{i,j+1}^{(2)} + 2P_{i,j+1}^{(3)} + 2\bar{P}_{i,j+1}^{(0)} - \frac{3}{2}\bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i,j+1}^{(3)} = & \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} a_1(6u_m-3)F_{i,j+1}^{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} a_1^2(6u_m-3)^2} \\
\bar{P}_{i,j+1}^{(4)} = & \bar{P}_{i,j}^{(4)} + \bar{P}_{i,j}^{(7)} + \bar{P}_{i,j}^{(8)} \\
\bar{P}_{i,j+1}^{(5)} = & \bar{P}_{i,j}^{(5)} + \bar{P}_{i,j}^{(9)} \\
\bar{P}_{i,j+1}^{(6)} = & \bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j+1}^{(7)} = & -3P_{i,j+1}^{(5)} - P_{i,j+1}^{(3)} + 4P_{i,j+1}^{(4)} + 4P_{i,j+1}^{(6)} \\
& -4P_{i,j+1}^{(2)} - 3\bar{P}_{i,j+1}^{(4)} - 2\bar{P}_{i,j+1}^{(5)} - \frac{3}{2}\bar{P}_{i,j+1}^{(6)} \\
\bar{P}_{i,j+1}^{(8)} = & 2P_{i,j+1}^{(5)} - 2P_{i,j+1}^{(3)} - 4P_{i,j+1}^{(6)} + 4P_{i,j+1}^{(2)} \\
& +2\bar{P}_{i,j+1}^{(4)} + 2\bar{P}_{i,j+1}^{(5)} + 2\bar{P}_{i,j+1}^{(6)} \\
\bar{P}_{i,j+1}^{(9)} = & 2P_{i,j+1}^{(5)} + 2P_{i,j+1}^{(3)} - 4P_{i,j+1}^{(4)} - \bar{P}_{i,j+1}^{(5)} - \frac{3}{2}\bar{P}_{i,j+1}^{(6)}
\end{cases}
\qquad (A.4)
$$

where

$$
\begin{aligned}
F_{i,j+1}^{m,n} =\,& 2a_1(-4P_{i,j+1}^{(2)} + 2P_{i,j+1}^{(3)} + 2\bar{P}_{i,j+1}^{(0)} + v_n 2\bar{P}_{i,j+1}^{(8)}) \\
& + 2a_2(\bar{P}_{i,j+1}^{(5)} + 3v_n \bar{P}_{i,j+1}^{(6)} + u_m \bar{P}_{i,j+1}^{(9)})
\end{aligned}
\qquad (A.5)
$$

For $S_{i+1,j}$ patch, $\bar{P}_{i+1,j}^{(k)}$ ($k = 0, 1, 2, 3, ..., 9$) are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i+1,j}^{(0)} = & \bar{P}_{i,j}^{(0)} + \bar{P}_{i,j}^{(4)} + \bar{P}_{i,j}^{(5)} + \bar{P}_{i,j}^{(6)} \\
\bar{P}_{i+1,j}^{(1)} = & \bar{P}_{i,j}^{(1)} + \bar{P}_{i,j}^{(7)} + \bar{P}_{i,j}^{(9)} \\
\bar{P}_{i+1,j}^{(2)} = & \bar{P}_{i,j}^{(2)} + \bar{P}_{i,j}^{(8)} \\
\bar{P}_{i+1,j}^{(3)} = & \bar{P}_{i,j}^{(3)} \\
\bar{P}_{i+1,j}^{(4)} = & 4P_{i+1,j}^{(8)} - 3\bar{P}_{i+1,j}^{(0)} - P_{i+1,j}^{(7)} + \frac{1}{2}\bar{P}_{i+1,j}^{(6)} \\
\bar{P}_{i+1,j}^{(5)} = & -4P_{i+1,j}^{(8)} + 2\bar{P}_{i+1,j}^{(0)} + 2P_{i+1,j}^{(7)} - \frac{3}{2}\bar{P}_{i+1,j}^{(6)} \\
\bar{P}_{i+1,j}^{(6)} = & \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} a_2(6v_n-3)F_{i+1,j}^{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} a_2^2(6v_n-3)^2} \\
\bar{P}_{i+1,j}^{(7)} = & -3P_{i+1,j}^{(5)} - P_{i+1,j}^{(7)} + 4P_{i+1,j}^{(6)} + 4P_{i+1,j}^{(4)} \\
& -4P_{i+1,j}^{(8)} - 3\bar{P}_{i+1,j}^{(1)} - 2\bar{P}_{i+1,j}^{(2)} - \frac{3}{2}\bar{P}_{i+1,j}^{(3)} \\
\bar{P}_{i+1,j}^{(8)} = & 2P_{i+1,j}^{(5)} + 2P_{i+1,j}^{(7)} - 4P_{i+1,j}^{(6)} - \bar{P}_{i+1,j}^{(2)} - \frac{3}{2}\bar{P}_{i+1,j}^{(3)} \\
\bar{P}_{i+1,j}^{(9)} = & 2P_{i+1,j}^{(5)} - 2P_{i+1,j}^{(7)} - 4P_{i+1,j}^{(4)} + 4P_{i+1,j}^{(8)} \\
& +2\bar{P}_{i+1,j}^{(1)} + 2\bar{P}_{i+1,j}^{(2)} + 2\bar{P}_{i+1,j}^{(3)}
\end{cases} \tag{A.6}
$$

where

$$
\begin{aligned}
F_{i,j+1}^{m,n} =& 2a_1(\bar{P}_{i+1,j}^{(2)} + 3u_m\bar{P}_{i+1,j}^{(3)} + v_n\bar{P}_{i+1,j}^{(8)}) \\
& + 2a_2(u_m\bar{P}_{i+1,j}^{(9)} - 4P_{i+1,j}^{(8)} + 2\bar{P}_{i+1,j}^{(0)} + 2P_{i+1,j}^{(7)})
\end{aligned} \tag{A.7}
$$

For $S_{i,j-1}$ patch, $\bar{P}_{i,j-1}^{(k)}$ ($k = 0, 1, 2, 3, ..., 9$) are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i,j-1}^{(0)} = & P_{i,j-1}^{(1)} \\
\bar{P}_{i,j-1}^{(1)} = & -3P_{i,j-1}^{(1)} + 4P_{i,j-1}^{(2)} - \bar{P}_{i,j}^{(0)} + \frac{1}{2}\bar{P}_{i,j-1}^{(3)} \\
\bar{P}_{i,j-1}^{(2)} = & 2P_{i,j-1}^{(1)} - 4P_{i,j-1}^{(2)} + 2\bar{P}_{i,j}^{(0)} - \frac{3}{2}\bar{P}_{i,j-1}^{(3)} \\
\bar{P}_{i,j-1}^{(3)} = & \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} a_1(6u_m-3)F_{i,j-1}^{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} a_1^2(6u_m-3)^2} \\
\bar{P}_{i,j-1}^{(4)} = & 4P_{i,j-1}^{(8)} - 3P_{i,j-1}^{(1)} - P_{i,j-1}^{(7)} + \frac{1}{2}\bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j-1}^{(5)} = & -4P_{i,j-1}^{(8)} + 2P_{i,j-1}^{(1)} + 2P_{i,j-1}^{(7)} - \frac{3}{2}\bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j-1}^{(6)} = & \bar{P}_{i,j}^{(6)} \\
\bar{P}_{i,j-1}^{(7)} = & 4P_{i,j-1}^{(6)} - 4P_{i,j-1}^{(2)} + 5P_{i,j-1}^{(1)} - 4P_{i,j-1}^{(8)} \\
& -P_{i,j-1}^{(7)} - 2\bar{P}_{i,j}^{(5)} - \frac{5}{2}\bar{P}_{i,j}^{(6)} - \bar{P}_{i,j}^{(4)} \\
\bar{P}_{i,j-1}^{(8)} = & -4P_{i,j-1}^{(6)} + 4P_{i,j-1}^{(2)} - 2P_{i,j-1}^{(1)} + 2P_{i,j-1}^{(7)} \\
& +2\bar{P}_{i,j}^{(5)} + 2\bar{P}_{i,j}^{(6)} + 2\bar{P}_{i,j}^{(4)} \\
\bar{P}_{i,j-1}^{(9)} = & 4P_{i,j-1}^{(8)} - 2P_{i,j-1}^{(7)} - 2P_{i,j-1}^{(1)} + \bar{P}_{i,j}^{(5)} + \frac{3}{2}\bar{P}_{i,j}^{(6)}
\end{cases} \tag{A.8}
$$

where

$$
\begin{aligned}
F_{i,j-1}^{m,n} =& 2a_1(2P_{i,j-1}^{(1)} - 4P_{i,j-1}^{(2)} + 2\bar{P}_{i,j}^{(0)} + v_n\bar{P}_{i,j-1}^{(8)}) \\
& + 2a_2(\bar{P}_{i,j-1}^{(5)} - 3v_n\bar{P}_{i,j-1}^{(6)} + u_m\bar{P}_{i,j-1}^{(9)})
\end{aligned} \tag{A.9}
$$

For $S_{i-1,j}$ patch, $\bar{P}_{i-1,j}^{(k)}$ $(k = 0, 1, 2, 3, ..., 9)$ are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i-1,j}^{(0)} = & P_{i-1,j}^{(1)} \\
\bar{P}_{i-1,j}^{(1)} = & 4P_{i-1,j}^{(2)} - 3P_{i-1,j}^{(1)} - P_{i-1,j}^{(3)} + \frac{1}{2}\bar{P}_{i,j}^{(3)} \\
\bar{P}_{i-1,j}^{(2)} = & -4P_{i-1,j}^{(2)} + 2P_{i-1,j}^{(1)} + 2P_{i-1,j}^{(3)} - \frac{3}{2}\bar{P}_{i,j}^{(3)} \\
\bar{P}_{i-1,j}^{(3)} = & \bar{P}_{i,j}^{(3)} \\
\bar{P}_{i-1,j}^{(4)} = & 4P_{i-1,j}^{(8)} - 3P_{i-1,j}^{(1)} + \frac{1}{2}\bar{P}_{i-1,j}^{(6)} - \bar{P}_{i,j}^{(0)} \\
\bar{P}_{i-1,j}^{(5)} = & -4P_{i-1,j}^{(8)} + 2P_{i-1,j}^{(1)} - \frac{3}{2}\bar{P}_{i-1,j}^{(6)} + 2\bar{P}_{i,j}^{(0)} \\
\bar{P}_{i-1,j}^{(6)} = & \dfrac{\sum_{m=1}^{M}\sum_{n=1}^{N} a_2(6v_n-3)F_{i-1,j}^{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} a_2^2(6v_n-3)^2} \\
\bar{P}_{i-1,j}^{(7)} = & 4P_{i-1,j}^{(4)} - 4P_{i-1,j}^{(8)} + 5P_{i-1,j}^{(1)} - 4P_{i-1,j}^{(2)} \\
& -P_{i-1,j}^{(3)} - 2\bar{P}_{i,j}^{(2)} - \frac{5}{2}\bar{P}_{i,j}^{(3)} - \bar{P}_{i,j}^{(1)} \\
\bar{P}_{i-1,j}^{(8)} = & -2P_{i-1,j}^{(1)} + 4P_{i-1,j}^{(2)} - 2P_{i-1,j}^{(3)} + \bar{P}_{i,j}^{(2)} \\
& +\frac{3}{2}\bar{P}_{i,j}^{(3)} \\
\bar{P}_{i-1,j}^{(9)} = & -4P_{i-1,j}^{(4)} + 4P_{i-1,j}^{(8)} - 2P_{i-1,j}^{(1)} + 2P_{i-1,j}^{(3)} \\
& +2\bar{P}_{i,j}^{(2)} + 2\bar{P}_{i,j}^{(3)} + 2\bar{P}_{i,j}^{(1)}
\end{cases}
\tag{A.10}
$$

where

$$
\begin{aligned}
F_{i-1,j}^{m,n} =& 2a_1(\bar{P}_{i-1,j}^{(2)} + 3u_m\bar{P}_{i-1,j}^{(3)} + v_n\bar{P}_{i-1,j}^{(8)}) \\
& + 2a_2(u_m\bar{P}_{i-1,j}^{(9)} - 4\bar{P}_{i-1,j}^{(8)} + 2P_{i-1,j}^{(1)} + 2\bar{P}_{i,j}^{(0)})
\end{aligned}
\tag{A.11}
$$

For $S_{i+1,j+1}$ patch, $\bar{P}_{i+1,j+1}^{(k)}$ $(k = 0, 1, 2, 3, ..., 9)$ are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i+1,j+1}^{(0)} = & \bar{P}_{i+1,j}^{(0)} + \bar{P}_{i+1,j}^{(1)} + \bar{P}_{i+1,j}^{(2)} + \bar{P}_{i+1,j}^{(3)} \\
\bar{P}_{i+1,j+1}^{(1)} = & \bar{P}_{i,j+1}^{(1)} + \bar{P}_{i,j+1}^{(7)} + \bar{P}_{i,j+1}^{(9)} \\
\bar{P}_{i+1,j+1}^{(2)} = & \bar{P}_{i,j+1}^{(2)} + \bar{P}_{i,j+1}^{(8)} \\
\bar{P}_{i+1,j+1}^{(3)} = & \bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i+1,j+1}^{(4)} = & \bar{P}_{i+1,j}^{(4)} + \bar{P}_{i+1,j}^{(7)} + \bar{P}_{i+1,j}^{(8)} \\
\bar{P}_{i+1,j+1}^{(5)} = & \bar{P}_{i+1,j}^{(5)} + \bar{P}_{i+1,j}^{(9)} \\
\bar{P}_{i+1,j+1}^{(6)} = & \bar{P}_{i+1,j}^{(6)} \\
\bar{P}_{i+1,j+1}^{(7)} = & -3P_{i+1,j+1}^{(5)} + 4P_{i+1,j+1}^{(6)} + 4P_{i+1,j+1}^{(4)} - 5\bar{P}_{i+1,j+1}^{(0)} \\
& -3\bar{P}_{i+1,j+1}^{(1)} - 2\bar{P}_{i+1,j+1}^{(2)} - \frac{3}{2}\bar{P}_{i+1,j+1}^{(3)} - 3\bar{P}_{i+1,j+1}^{(4)} \\
& -2\bar{P}_{i+1,j+1}^{(5)} - \frac{3}{2}\bar{P}_{i+1,j+1}^{(6)} \\
\bar{P}_{i+1,j+1}^{(8)} = & 2P_{i+1,j+1}^{(5)} - 4P_{i+1,j+1}^{(6)} + 2\bar{P}_{i+1,j+1}^{(0)} - \bar{P}_{i+1,j+1}^{(2)} \\
& -\frac{3}{2}\bar{P}_{i+1,j+1}^{(3)} + 2\bar{P}_{i+1,j+1}^{(4)} + 2\bar{P}_{i+1,j+1}^{(5)} + 2\bar{P}_{i+1,j+1}^{(6)} \\
\bar{P}_{i+1,j+1}^{(9)} = & 2P_{i+1,j+1}^{(5)} - 4P_{i+1,j+1}^{(4)} + 2\bar{P}_{i+1,j+1}^{(0)} + 2\bar{P}_{i+1,j+1}^{(1)} \\
& +2\bar{P}_{i+1,j+1}^{(2)} + 2\bar{P}_{i+1,j+1}^{(3)} - \bar{P}_{i+1,j+1}^{(5)} - \frac{3}{2}\bar{P}_{i+1,j+1}^{(6)}
\end{cases}
\tag{A.12}
$$

For $S_{i-1,j-1}$ patch, $\bar{P}_{i-1,j-1}^{(k)}$ $(k = 0,1,2,3,...,9)$ are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i-1,j-1}^{(0)} = & P_{i-1,j-1}^{(1)} \\
\bar{P}_{i-1,j-1}^{(1)} = & 4P_{i-1,j-1}^{(2)} - 3P_{i-1,j-1}^{(1)} + \frac{1}{2}\bar{P}_{i,j-1}^{(3)} - \bar{P}_{i-1,j}^{(0)} \\
\bar{P}_{i-1,j-1}^{(2)} = & -4P_{i-1,j-1}^{(2)} + 2P_{i-1,j-1}^{(1)} - \frac{3}{2}\bar{P}_{i,j-1}^{(3)} + 2\bar{P}_{i-1,j}^{(0)} \\
\bar{P}_{i-1,j-1}^{(3)} = & \bar{P}_{i,j-1}^{(3)} \\
\bar{P}_{i-1,j-1}^{(4)} = & 4P_{i-1,j-1}^{(8)} - 3P_{i-1,j-1}^{(1)} + \frac{1}{2}\bar{P}_{i-1,j}^{(6)} - \bar{P}_{i,j-1}^{(0)} \\
\bar{P}_{i-1,j-1}^{(5)} = & -4P_{i-1,j-1}^{(8)} + 2P_{i-1,j-1}^{(1)} - \frac{3}{2}\bar{P}_{i-1,j}^{(6)} + 2\bar{P}_{i,j-1}^{(0)} \\
\bar{P}_{i-1,j-1}^{(6)} = & \bar{P}_{i-1,j}^{(6)} \\
\bar{P}_{i-1,j-1}^{(7)} = & 5P_{i-1,j-1}^{(1)} - 4P_{i-1,j-1}^{(2)} - 4P_{i-1,j-1}^{(8)} + 2\bar{P}_{i-1,j}^{(0)} \\
& +\bar{P}_{i-1,j}^{(4)} - \frac{1}{2}\bar{P}_{i-1,j}^{(6)} + \bar{P}_{i,j-1}^{(0)} - \bar{P}_{i,j-1}^{(2)} - \frac{3}{2}\bar{P}_{i,j-1}^{(3)} \\
\bar{P}_{i-1,j-1}^{(8)} = & -2P_{i-1,j-1}^{(1)} + 4P_{i-1,j-1}^{(2)} - 2\bar{P}_{i-1,j}^{(0)} \\
& +\bar{P}_{i,j-1}^{(2)} + \frac{3}{2}\bar{P}_{i,j-1}^{(3)} \\
\bar{P}_{i-1,j-1}^{(9)} = & -2P_{i-1,j-1}^{(1)} + 4P_{i-1,j-1}^{(8)} - 2\bar{P}_{i,j-1}^{(0)} \\
& +\bar{P}_{i-1,j}^{(5)} + \frac{3}{2}\bar{P}_{i-1,j}^{(6)}
\end{cases}
\tag{A.13}
$$

For $S_{i-1,j+1}$ patch, $\bar{P}_{i-1,j+1}^{(k)}$ $(k = 0,1,2,3,...,9)$ are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}_{i-1,j+1}^{(0)} = & \bar{P}_{i-1,j}^{(0)} + \bar{P}_{i-1,j}^{(1)} + \bar{P}_{i-1,j}^{(2)} + \bar{P}_{i-1,j}^{(3)} \\
\bar{P}_{i-1,j+1}^{(1)} = & 4P_{i-1,j+1}^{(2)} - P_{i-1,j+1}^{(3)} - 3\bar{P}_{i-1,j+1}^{(0)} + \frac{1}{2}\bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i-1,j+1}^{(2)} = & -4P_{i-1,j+1}^{(2)} + 2P_{i-1,j+1}^{(3)} + 2\bar{P}_{i-1,j+1}^{(0)} - \frac{3}{2}\bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i-1,j+1}^{(3)} = & \bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i-1,j+1}^{(4)} = & \bar{P}_{i-1,j}^{(4)} + \bar{P}_{i-1,j}^{(7)} + \bar{P}_{i-1,j}^{(8)} \\
\bar{P}_{i-1,j+1}^{(5)} = & \bar{P}_{i-1,j}^{(5)} + \bar{P}_{i-1,j}^{(9)} \\
\bar{P}_{i-1,j+1}^{(6)} = & \bar{P}_{i-1,j}^{(6)} \\
\bar{P}_{i-1,j+1}^{(7)} = & 4P_{i-1,j+1}^{(4)} - P_{i-1,j+1}^{(3)} - 4P_{i-1,j+1}^{(2)} + \bar{P}_{i-1,j+1}^{(0)} \\
& -2\bar{P}_{i-1,j+1}^{(4)} - \bar{P}_{i-1,j+1}^{(5)} - \frac{1}{2}\bar{P}_{i-1,j+1}^{(6)} - 2\bar{P}_{i,j+1}^{(2)} \\
& -\frac{5}{2}\bar{P}_{i,j+1}^{(3)} - \bar{P}_{i,j+1}^{(1)} \\
\bar{P}_{i-1,j+1}^{(8)} = & -2P_{i-1,j+1}^{(3)} + 4P_{i-1,j+1}^{(2)} - 2\bar{P}_{i-1,j+1}^{(0)} \\
& +\bar{P}_{i,j+1}^{(2)} + \frac{3}{2}\bar{P}_{i,j+1}^{(3)} \\
\bar{P}_{i-1,j+1}^{(9)} = & -4P_{i-1,j+1}^{(4)} + 2P_{i-1,j+1}^{(3)} + 2\bar{P}_{i-1,j+1}^{(0)} \\
& +2\bar{P}_{i-1,j+1}^{(4)} + \bar{P}_{i-1,j+1}^{(5)} + \frac{1}{2}\bar{P}_{i-1,j+1}^{(6)} \\
& +2\bar{P}_{i,j+1}^{(1)} + 2\bar{P}_{i,j+1}^{(2)} + 2\bar{P}_{i,j+1}^{(3)}
\end{cases}
\tag{A.14}
$$

For $S_{i+1,j-1}$ patch, $\bar{P}^{(k)}_{i+1,j-1}$ $(k = 0, 1, 2, 3, ..., 9)$ are explicitly determined by the following analytical formulae

$$
\begin{cases}
\bar{P}^{(0)}_{i+1,j-1} = & \bar{P}^{(0)}_{i,j-1} + \bar{P}^{(4)}_{i,j-1} + \bar{P}^{(5)}_{i,j-1} + \bar{P}^{(6)}_{i,j-1} \\
\bar{P}^{(1)}_{i+1,j-1} = & \bar{P}^{(1)}_{i,j-1} + \bar{P}^{(7)}_{i,j-1} + \bar{P}^{(9)}_{i,j-1} \\
\bar{P}^{(2)}_{i+1,j-1} = & \bar{P}^{(2)}_{i,j-1} + \bar{P}^{(8)}_{i,j-1} \\
\bar{P}^{(3)}_{i+1,j-1} = & \bar{P}^{(3)}_{i,j-1} \\
\bar{P}^{(4)}_{i+1,j-1} = & 4P^{(8)}_{i+1,j-1} - P^{(7)}_{i+1,j-1} - 3\bar{P}^{(0)}_{i+1,j-1} + \frac{1}{2}\bar{P}^{(6)}_{i+1,j} \\
\bar{P}^{(5)}_{i+1,j-1} = & -4P^{(8)}_{i+1,j-1} + 2P^{(7)}_{i+1,j-1} + 2\bar{P}^{(0)}_{i+1,j-1} - \frac{3}{2}\bar{P}^{(6)}_{i+1,j} \\
\bar{P}^{(6)}_{i+1,j-1} = & \bar{P}^{(6)}_{i+1,j} \\
\bar{P}^{(7)}_{i+1,j-1} = & -4P^{(8)}_{i+1,j-1} - P^{(7)}_{i+1,j-1} + 4P^{(6)}_{i+1,j-1} + \bar{P}^{(0)}_{i+1,j-1} \\
& -2\bar{P}^{(1)}_{i+1,j-1} - \bar{P}^{(2)}_{i+1,j-1} - \frac{1}{2}\bar{P}^{(3)}_{i+1,j-1} - \bar{P}^{(4)}_{i+1,j} \\
& -2\bar{P}^{(5)}_{i+1,j} - \frac{5}{2}\bar{P}^{(6)}_{i+1,j} \\
\bar{P}^{(8)}_{i+1,j-1} = & 2P^{(7)}_{i+1,j-1} - 4P^{(6)}_{i+1,j-1} + 2\bar{P}^{(0)}_{i+1,j-1} \\
& +2\bar{P}^{(1)}_{i+1,j-1} + \bar{P}^{(2)}_{i+1,j-1} + \frac{1}{2}\bar{P}^{(3)}_{i+1,j-1} \\
& +2\bar{P}^{(4)}_{i+1,j} + 2\bar{P}^{(5)}_{i+1,j} + 2\bar{P}^{(6)}_{i+1,j} \\
\bar{P}^{(9)}_{i+1,j-1} = & 4P^{(8)}_{i+1,j-1} - 2P^{(7)}_{i+1,j-1} - 2\bar{P}^{(0)}_{i+1,j-1} \\
& +\bar{P}^{(5)}_{i+1,j} + \frac{3}{2}\bar{P}^{(6)}_{i+1,j}
\end{cases} \tag{A.15}
$$