Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 188 (2003) 75–99

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A computational model for organism growth based on surface mesh generation

C.H. Leung, M. Berzins *

School of Computing, University of Leeds, Leeds LS2 9JT, UK

Received 5 November 2002; received in revised form 29 January 2003; accepted 30 January 2003

Abstract

The generation of curvilinear coordinate meshes has been a powerful tool in computational fluid dynamics calculation in the computational modelling of the fluid flows around complex bodies, such as an airfoil or a complete aircraft. This same technique may be applied to many other computational models. In this work the approach is used as part of a computational model to generate simple geometries associated with biological forms or organisms. The model adopted was first proposed by Cummings and simulates morphogenesis in terms of the geometrical changes occurring during the growth and development of simple organisms. This model depends on the generation of a curvilinear coordinate mesh on the surface of an organism. Previous work has concentrated on the model and its use in generating axisymmetric shapes that are simple models of elementary 'organisms'. In this work we describe how the model may be extended to geometrical symmetry breaking. This paper describes the methodology of this extension and demonstrates it in the simulation of tentacle growth. The resulting computational technique makes it possible to link models of cell bio-chemistry and surface deformation.

Keywords: Mesh generation; Parabolic equations; Developmental biology; Pattern formation

1. Introduction

In the model of morphogenesis proposed by Cummings [7–9,11], an organism that initially takes the form of a sphere evolves into a certain geometry. The growth of the organism is modelled and is parametrised by the surface area of its epithelial layer. As the area increases, a pair of chemicals called morphogens with concentrations denoted by $\mathbf{m} = (m_1, m_2)$, interact in this epithelial layer, forming a spatial pattern of morphogen concentrations. In an approach dating back to Turing [31], the biochemical reactions are governed by a system of RD (reaction diffusion) equations

[*] Corresponding author.
E-mail addresses: hang@comp.leeds.ac.uk (C.H. Leung), martin@comp.leeds.ac.uk (M. Berzins).

0021-9991/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00153-0

$$\frac{\partial \mathbf{m}}{\partial t} = \Delta_2 \mathbf{m} + F(\mathbf{m}). \tag{1}$$

Since Turing's work, morphogens have been used in modelling pattern formation and the regeneration of body parts in hydra [20,24,25]; the pre-patterning of animal coat markings [23,26] and the outgrowth of a vertebrate limb bud [12]. In (1), Δ_2 denotes the Laplace–Beltrami operator that governs the diffusion on the surface and is characterised by the surface geometry. The Laplace–Beltrami operator has the form

$$\Delta_2 = \frac{1}{g}\left\{\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}\right\}, \tag{2}$$

where surface geometry is specified by a quantity g that is related to the metric tensor components (explained below). The assumption is that any parametrisation of the surface (and hence its growth) depends on the local concentration of the morphogens. One such parametrisation is in terms of surface curvatures, such as the Gauss and the Mean curvatures [1] (see [4–6]). With the specification of the Gauss and Mean curvatures $K(\mathbf{m})$ and $H(\mathbf{m})$ as functions of $\mathbf{m}$, this morphogen pattern then alters the surface geometry through the modification of $K(\mathbf{m})$ and $H(\mathbf{m})$. Since a new geometry is formed, the operator Δ_2 therefore is changed accordingly, thus leading to a new morphogen pattern. The cycle then repeats thus forming a morphogenetic feedback loop. The Laplace–Beltrami operator Δ_2 is specified on abstract coordinates (ξ_1, ξ_2) which preserve the form of differential arc length and area under conformal transformation [6]. This abstract coordinate is intrinsic to the particular surface geometry.

The formation of this intrinsic coordinate can be envisaged as mapping a planar mesh, i.e., a square mesh, onto the non-planar surface such that the mesh remains *orthogonal* as well as *isothermal* on the surface. Suppose a planar mesh defined by $[0 < \xi_1 < a] \times [0 < \xi_2 < b]$ is mapped onto a non-planar surface parametrised by $\mathbf{r} = \mathbf{r}(x(\xi_1, \xi_2), y(\xi_1, \xi_2), z(\xi_1, \xi_2))$, an *orthogonal* mesh is then defined as one in which the four corners of every cells on the mesh remain at right angle, i.e., the tangents in the two perpendicular directions, $\partial\mathbf{r}/\partial\xi_1$ and $\partial\mathbf{r}/\partial\xi_2$, intersect at a right angle everywhere, i.e., their dot product vanishes,

$$\frac{\partial \mathbf{r}}{\partial \xi_1}\cdot\frac{\partial \mathbf{r}}{\partial \xi_2} = \left|\frac{\partial \mathbf{r}}{\partial \xi_1}\right|\left|\frac{\partial \mathbf{r}}{\partial \xi_2}\right|\cos\theta = g_{12} = g_{21} = 0 \quad \text{with } \theta = \frac{\pi}{2}, \tag{3}$$

where θ is the angle between the two adjacent sides of a cell; and an *isothermal* mesh is one in which the grid-lines are distributed such that the rate of change of arc length in the two directions are everywhere equal, i.e., the magnitudes of the two perpendicular tangents, $\partial\mathbf{r}/\partial\xi_1$ and $\partial\mathbf{r}/\partial\xi_2$, are equal to

$$\frac{\partial \mathbf{r}}{\partial \xi_1}\cdot\frac{\partial \mathbf{r}}{\partial \xi_1} = g_{11} = \frac{\partial \mathbf{r}}{\partial \xi_2}\cdot\frac{\partial \mathbf{r}}{\partial \xi_2} = g_{22} = g. \tag{4}$$

Here g_{11}, g_{12}, g_{21} and g_{22} denote the four metric tensor components. The mapping of a square mesh onto a general curved surface is illustrated in Fig. 1, with the two surface base vectors denoted by $\partial\mathbf{r}/\partial\xi_1$ and $\partial\mathbf{r}/\partial\xi_2$, respectively. The arc length ds on the surface is determined by the metric g. On a general surface with a mesh defined by the tangent vectors $\partial\mathbf{r}/\partial\xi_1$ and $\partial\mathbf{r}/\partial\xi_2$, d$s$ is given by

$$\mathrm{d}s^2 = \mathrm{d}\mathbf{r}\cdot\mathrm{d}\mathbf{r} = \left(\frac{\partial \mathbf{r}}{\partial \xi_1}\mathrm{d}\xi_1 + \frac{\partial \mathbf{r}}{\partial \xi_2}\mathrm{d}\xi_2\right)^2 = g_{11}\mathrm{d}\xi_1^2 + 2g_{12}\mathrm{d}\xi_1\,\mathrm{d}\xi_2 + g_{22}\mathrm{d}\xi_2^2. \tag{5}$$

When this mesh is orthogonal ($g_{12} = 0$) and isothermal ($g_{11} = g_{22} = g$), ds becomes

[1] A qualitative description of the Gauss and Mean curvatures and the associated surfaces are briefly discussed in Appendix A, a more comprehensive description and the mathematical treatment can be found in [22].

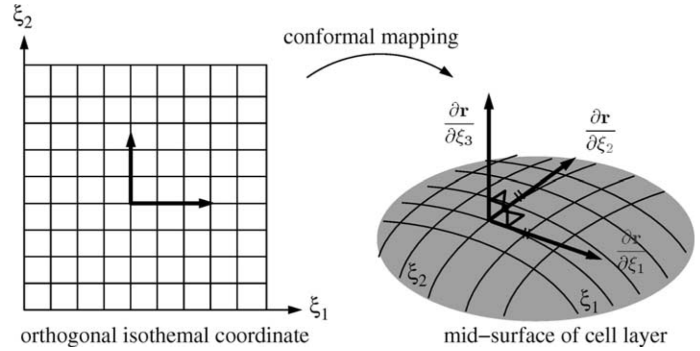
ξ_2
conformal mapping
$\frac{\partial \mathbf{r}}{\partial \xi_3}$
$\frac{\partial \mathbf{r}}{\partial \xi_2}$
$\frac{\partial \mathbf{r}}{\partial \xi_1}$
ξ_2
ξ_1
ξ_1
orthogonal isothemal coordinate
mid–surface of cell layer

Fig. 1. The three base vectors of a body-fitted curvilinear coordinate system on an arbitrary surface. The surface lies on the $\xi_3 =$ constant plane in the ξ_1, ξ_2, ξ_3 space.

$$ds^2 = g\left(d\xi_1^2 + d\xi_2^2\right). \tag{6}$$

Provided K is a known function, the metric tensor g can be computed by solving the Gauss equation

$$\left\{\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}\right\} \ln g + 2gK = 0. \tag{7}$$

The geometry in the Euclidean space E^3 can be reconstructed using the knowledge of g. For this reconstruction to be tractable, Cummings assumes an axisymmetric form for the organisms. Under this assumption, we have the following:

$$\rho = \sqrt{g} \quad \text{and} \quad z = \int_{\xi_1=0}^{\xi_1=a} \operatorname{sign}(H)\sqrt{\rho^2 - \left(\frac{d\rho}{d\xi_1}\right)^2}\, d\xi_1, \tag{8}$$

where the vertical distance z is expressed as an integral function of the distance from the axis of symmetry ρ.

Fig. 2 provides a summary of the morphogenetic feedback loop of Cummings' model.

The model formulation at the current state only deals with axisymmetric geometries, such as a sphere, in which one of the directions of the orthogonal isothermal mesh, ξ_2, say, is identical to the azimuthal angle. (Thus, it has the range of $[0 < \xi_2 < 2\pi]$.) As a result, the metric tensor has no dependence upon ξ_2.

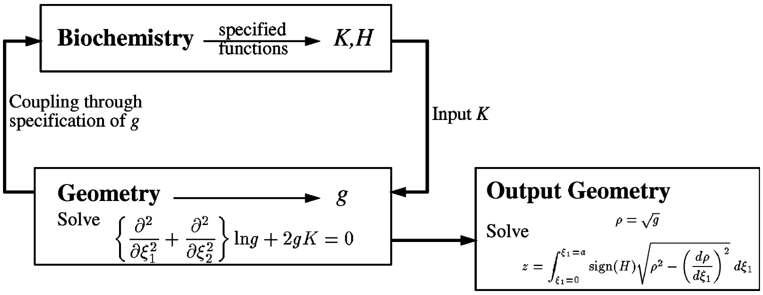
Biochemistry
specified functions
K,H
Coupling through specification of g
Input K
Geometry
g
Solve
$\left\{\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}\right\} \ln g + 2gK = 0$
Output Geometry
Solve
$\rho = \sqrt{g}$
$z = \int_{\xi_1=0}^{\xi_1=a} \operatorname{sign}(H)\sqrt{\rho^2 - \left(\frac{d\rho}{d\xi_1}\right)^2}\, d\xi_1$

Fig. 2. An overview of Cummings model. The coupling between biochemistry and geometry is achieved via the input of K (Gauss curvature) into the Gauss equation in the 'Geometry' block and the input of g (metric) into pattern generation in the 'Biochemistry' block.

Nevertheless, when the geometry ceases to remain axisymmetric, the metric tensor g can be a function of both ξ_1 and ξ_2. Thus the above integral equation that relates ρ and z no longer applies and one has to seek a more general solution strategy.

The purpose of this paper is to: first introduce the mathematical formulation of an alternative algorithm and methodology that allows the construction of a general intrinsic orthogonal isothermal mesh (ξ_1, ξ_2) that simulates non-axisymmetric geometries; then discuss the reconstruction of the geometry in E^3 in (ξ_1, ξ_2). The algorithm is then demonstrated to reproduce examples of non-axisymmetric geometries in E^3 from their K and H specifications in (ξ_1, ξ_2).

2. Evolving surface and fluid flow over surface

The technique of mapping a mesh onto a surface has been used in computational fluid dynamics to calculate flow properties around rigid body such as an airfoil (see Fig. 3). The idea is that rigid body such as an airfoil often has shape that does not coincide with any grid-line in a conventional Cartesian mesh. Numerical accuracy is then lost during computations on such a mesh. A transformation is then devised to map a mesh onto the outline or surface of the body such that the outline or the surface of the body coincides with a grid-line or coordinate plane. Such a mesh is called the body-fitted-coordinate, or curvilinear coordinate and the technique is called mesh generation [1,3,15,19,29].

With this concept in mind, the mapping of the orthogonal isothermal mesh (ξ_1, ξ_2) to a arbitrary curved (organism) surface is equivalent to the generation of a curvilinear coordinate system in E^3, i.e. (x_1, x_2, x_3). The arbitrary surface lies on a constant value of one of the coordinates, i.e., one seeks the mapping from $(\xi_1, \xi_2, \xi_3 = \text{constant})$ into (x_1, x_2, x_3). This re-introduces the surface normal to the $\xi_3 = $ constant plane. This surface normal $\mathbf{n}$ and the remaining two surface tangents form the base vectors in the curvilinear coordinate system

$$\mathbf{n} = \frac{\partial \mathbf{r}}{\partial \xi_3} = \left(\frac{\partial \mathbf{r}}{\partial \xi_1} \times \frac{\partial \mathbf{r}}{\partial \xi_2} \right) \Bigg/ \left| \frac{\partial \mathbf{r}}{\partial \xi_1} \times \frac{\partial \mathbf{r}}{\partial \xi_2} \right|, \quad \frac{\partial \mathbf{r}}{\partial \xi_1} \quad \text{and} \quad \frac{\partial \mathbf{r}}{\partial \xi_2}. \tag{9}$$

The curvilinear coordinate system and the corresponding base vectors are illustrated in Fig. 1. This technique may then, as will be shown below, be applied in the context of the approach of Cummings by generating curvilinear coordinates on an evolving surface.

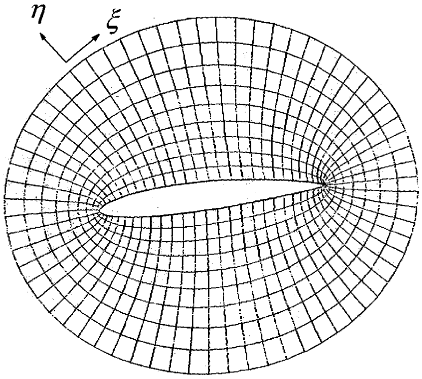

Fig. 3. The mapping of a body-fitted curvilinear coordinate system ξ, η onto a region surrounding a Karman–Trefftz airfoil [29]. The grid-lines parallel with the airfoil are the $\eta = $ constant lines whereas the grid-lines perpendicular to the airfoil are the $\xi = $ constant lines.

3. Curvilinear coordinate generation

In the generation of a curvilinear mesh on a surface, one solves for the mapping $M : \mathbf{r}(x_1, x_2, x_3) \rightarrow (\xi_1, \xi_2, \xi_3)$ on a close boundary. This is a boundary value problem of the elliptic type (see [29]),

$$\nabla^2 \xi_i = P^i, \quad i = 1, 2, \tag{10}$$

where $\nabla^2 = \partial^2/\partial x_1^2 + \partial^2/\partial x_2^2 + \partial^2/\partial x_3^2$ is the Laplacian operator. (The Laplace operator is used when it is applied to the case of planar mesh generation, whereas the Laplace–Beltrami operator

$$\Delta_2 = \frac{\partial}{\partial \xi_1}\left(\frac{g_{22}}{\sqrt{G_3}}\frac{\partial}{\partial \xi_1} - \frac{g_{12}}{\sqrt{G_3}}\frac{\partial}{\partial \xi_2}\right) + \frac{\partial}{\partial \xi_2}\left(\frac{g_{11}}{\sqrt{G_3}}\frac{\partial}{\partial \xi_2} - \frac{g_{12}}{\sqrt{G_3}}\frac{\partial}{\partial \xi_1}\right), \tag{11}$$

with

$$G_3 = g_{11}g_{22} - g_{12}^2 \tag{12}$$

is used instead in the case of general non-planar surface mesh generation. The quantity G_3 is the determinant of the covariant metric tensor on the ξ_3 plane, and is equal to the square of a differential area on the mesh.) The non-homogeneous term P^i is the mesh-density-control function, where $P^i = 0$ gives a uniform mesh. The solution of an elliptic PDE such as (10) satisfies the maximum principle [30] thus ensuring a one-to-one relationship in the mapping. Nevertheless, during morphogenesis, the surface definition (x_1, x_2, x_3) changes while it is mapped onto a fixed domain in the curvilinear coordinate, (ξ_1, ξ_2, ξ_3). The alternative is to seek the solution of the mapping $M^* : (\xi_1, \xi_2, \xi_3) \rightarrow \mathbf{r}(x_1, x_2, x_3)$ instead.

3.1. Mesh generation equation

The general mesh generation equation defining the surface $\mathbf{r}(x_1, x_2, x_3)$ using coordinates ξ_1, ξ_2 is given by

$$g_{22}\frac{\partial^2 \mathbf{r}}{\partial \xi_1^2} - 2g_{12}\frac{\partial^2 \mathbf{r}}{\partial \xi_1\,\partial \xi_2} + g_{11}\frac{\partial^2 \mathbf{r}}{\partial \xi_2^2} + G_3\left[\Delta_2\xi_1\frac{\partial \mathbf{r}}{\partial \xi_1} + \Delta_2\xi_2\frac{\partial \mathbf{r}}{\partial \xi_2}\right] = G_3\mathbf{n}(k_{\mathrm{I}} + k_{\mathrm{II}}), \tag{13}$$

where $k_{\mathrm{I}}, k_{\mathrm{II}}$ are the principal curvatures (see Appendix A) and Δ_2 is the Laplace–Beltrami operator given in (11). The derivation of this equation can be found in [33–35,37] (also see Appendix B). The normal $\mathbf{n} = (X_1, X_2, X_3)$ is found from the vector product in (9) to give

$$X_1 = J_1/\sqrt{G_3}, \quad X_2 = J_2/\sqrt{G_3}, \quad X_3 = J_3/\sqrt{G_3}, \tag{14}$$

where

$$J_1 = \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_2}{\partial \xi_2}\frac{\partial x_3}{\partial \xi_1}, \quad J_2 = \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_3}{\partial \xi_1} - \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2}, \quad J_3 = \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_1}.$$

The Gauss and the Mean curvatures, K and H, are related to the two principal curvatures by the definitions

$$K \equiv k_{\mathrm{I}}k_{\mathrm{II}}, \quad H \equiv \frac{k_{\mathrm{I}} + k_{\mathrm{II}}}{2}. \tag{15}$$

In the case of the model of Cummings the Gauss and Mean curvatures, K and H, are assumed to be defined by the morphogens and so are specified by another part of the model.

3.2. Time-dependent mesh generation

In the original Cummings model, the effects of the difference in time scale between growth and morphogen distribution were used to argue that as biochemistry is 'fast' in comparison with growth, the surface geometry and growth could be viewed as developing as a series of steady states. In terms of computation, one solves a set of boundary value problems. Although this approach is based on sound physical reasoning and one can use the previous solutions as estimates to obtain the next one, it has by experiment been found to be computationally advantageous to include transient effects by using the time-dependent form of the mesh generation equation.

The time-dependent mesh generation equation can be derived from (13) as shown in [37]. When a surface grid changes with time due to the change of the physical boundary, (10) is replaced by a parabolic equation of the following form:

$$\Delta_2 \xi_\delta = P^\delta + \frac{c}{G_3}\frac{\partial \xi_\delta}{\partial t}, \quad \text{where } \delta = 1,2, \tag{16}$$

where P^δ is an arbitrary prescribed mesh control function and c is a time scale. This is suggested in [21,37] as the simplest form that includes a transient term applied to the curvilinear coordinates and is equivalent to its elliptic counterpart.

Since the curvilinear variable ξ_3 is always constant on the surface of the body, its time derivative vanishes, then the rate of the change of the surface $\partial \mathbf{r}/\partial t$ is related to the rate of change of the remaining two curvilinear variables ξ_1, ξ_2 by the relation (see [21,32,36])

$$\frac{\partial \mathbf{r}}{\partial t} + \frac{\partial \mathbf{r}}{\partial \xi_1}\frac{\partial \xi_1}{\partial t} + \frac{\partial \mathbf{r}}{\partial \xi_2}\frac{\partial \xi_2}{\partial t} = 0. \tag{17}$$

Upon substitution of (16) and (17) into (13), one obtains

$$\frac{\partial \mathbf{r}}{\partial \sigma} = \mathscr{L}\mathbf{r} - \mathbf{n}G_3(k_\mathrm{I} + k_\mathrm{II}), \tag{18}$$

where $\sigma = t/c$ is the dimensionless time, the operator $\mathscr{L}$ is given by

$$\mathscr{L} = \left\{ g_{22}\frac{\partial^2}{\partial \xi_1^2} - 2g_{12}\frac{\partial^2}{\partial \xi_1\,\partial \xi_2} + g_{11}\frac{\partial^2}{\partial \xi_2^2} + P^1\frac{\partial}{\partial \xi_1} + P^2\frac{\partial}{\partial \xi_2} \right\}. \tag{19}$$

In (18), the control functions P^1 and P^2 are used to obtain the desired local mesh density on $\xi_1\xi_2$ space. Thus, the two control functions can be set to zero if a uniform mesh is desired [30].

In time-dependent grid generation, both K and H are functions of time so that the curvilinear mesh evolves as K and H changes. Alternatively, K and H may depend on a time-dependent variable such as the morphogens, as is the case here, which has to be solved simultaneously with the grid generator.

3.3. Incorporation of orthogonal isothermal criteria

The mesh generation technique developed in the previous sections can now be applied to generate the curvilinear coordinate on the organism's surface geometry in Cummings model. In this model, the surface coordinates are both orthogonal and isothermal, therefore upon applying the two conditions, $g = g_{11} = g_{22}$ and $g_{12} = 0$, to (18) and using the derived relation $G_3 = g^2$, this mesh generation equation reduces to

$$\frac{\partial \mathbf{r}}{\partial \sigma} = g\left\{\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}\right\}\mathbf{r} - 2\mathbf{n}gH, \tag{20}$$

where

$$g = \left(\frac{\partial x_1}{\partial \xi_1}\right)^2 + \left(\frac{\partial x_2}{\partial \xi_1}\right)^2 + \left(\frac{\partial x_3}{\partial \xi_1}\right)^2 = \left(\frac{\partial x_1}{\partial \xi_2}\right)^2 + \left(\frac{\partial x_2}{\partial \xi_2}\right)^2 + \left(\frac{\partial x_3}{\partial \xi_2}\right)^2.$$

This method of mapping **r** onto ξ_1, ξ_2 by explicitly specifying $g = g_{11} = g_{22}$. Note that in [27], the isothermal condition $g = g_{11} = g_{22}$ was interpreted as

$$f(\xi_1, \xi_2) = \sqrt{g_{22}/g_{11}}. \tag{21}$$

In addition, $g_{12} = 0$ in the mesh generation equation is known as the *strong constraint method* in [27]. This is particularly used when the physical geometry is yet to be determined and has to be computed as part of the solution.

The boundary orthogonality is achieved by using Neumann condition. In-so-doing, the control functions Ps in (19) remain as zero. [2] This method allows grid point to slide along the boundary until orthogonality property is achieved, (see [17]). To apply Neumann condition along the boundary $\xi_1 = l$, the formula $g_{12} = 0$, i.e.,

$$\frac{\partial x_1}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} + \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} + \frac{\partial x_3}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} = 0, \tag{22}$$

is used in conjunction with

$$\frac{\partial x_3}{\partial \xi_1} = R\,\mathrm{sech}^2\xi_1 \quad \text{and} \quad \frac{\partial x_3}{\partial \xi_2} = 0. \tag{23}$$

The isothermal criterion is then satisfied by using the formula $g_{11} = g_{22}$ (see (20)).

The geometry near the region $\xi_1 = -l$ is assumed to be a sphere, thus the boundary conditions are given by

$$x_1 = R\,\mathrm{sech}\xi_1 \cos \xi_2, \quad x_2 = R\,\mathrm{sech}\xi_1 \sin \xi_2, \quad x_3 = R \tanh \xi_1, \tag{24}$$

which is the Mercator's transformation [13] (also see Appendix C). The radius R can be obtained from $K = 1/R^2$ or $H = 1/R$. This is chosen because the mapping of a sphere solution already satisfies the orthogonal and isothermal criteria. Along the boundaries $\xi_2 = 0, 2\pi$, periodic conditions are incorporated except for the value of x_2, which is found from

$$x_2(\xi_2 = 0) = x_2(\xi_2 = 2\pi) - 2\pi. \tag{25}$$

This is chosen simply to enforce the continuity of the solution across the boundaries $\xi_2 = 0, 2\pi$.

3.4. Time-dependent version of Cummings model

The version of Cummings model used in this work can now be stated. The key aspect of Cummings model that remains unchanged is that the biochemical reactions on the surface forms a morphogen

[2] Alternatively, if Dirichlet condition is used, non-zero control functions Ps have to be used in order to safeguard boundary orthogonality [17].

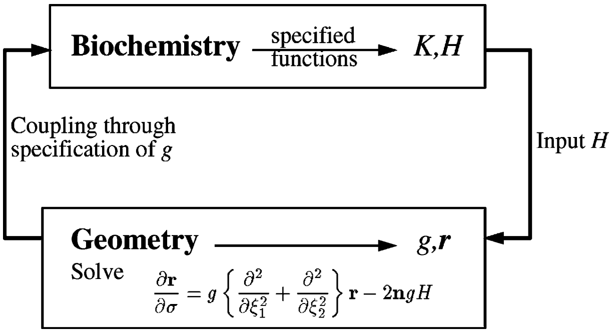
Biochemistry
specified functions
K,H
Coupling through specification of g
Input H
Geometry
g,r
Solve $\frac{\partial \mathbf{r}}{\partial \sigma} = g\left\{\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}\right\}\mathbf{r} - 2\mathbf{n}gH$

Fig. 4. An overview of an alternative algorithm of Cummings model. The coupling between 'Biochemistry' and 'Geometry' is achieved via the input of H (Mean curvature) into the mesh generation in the 'Geometry' block and the g (metric) into the pattern generation in the 'Biochemistry' block.

pattern of which the local concentration is used to define local surface curvatures (see [10,11]). The surface geometry is obtained by solving the mesh generation equation, i.e. (20), to obtain the Euclidean geometry **r** directly. The metric tensor g can be computed from **r** as a posterior calculation. Note that it is the Mean curvature H that is used as the input to the geometry specification in (20). In this version, the growth continues with time, one therefore solves a parabolic IBVP (initial boundary value problem). In contrast, the metric tensor g is solved for directly as a solution of the Gauss equation, i.e. (7). The Euclidean geometry **r** is then reconstructed using (8), i.e., it is obtained from posterior calculation. Moreover, from (7), K is used as the input for geometry specification in the computation of g. The growth in the original is parametrised by the surface area and it evolves in a series of steady states, i.e., one solves a elliptic BVP (boundary value problem). In both versions, g has to be computed, either as a posterior calculation in solving the mesh generation equation or directly from the Gauss equation, for the use of the geometry definition in the Laplace–Beltrami operator in the computation of morphogens' concentration (see (1)). The overall alternative algorithm is illustrated in Fig. 4 (compared with the original one illustrated in Fig. 2).

4. Artificial curvatures generation

In the foregoing sections, an algorithm has been derived which can generate asymmetric geometries from known curvature fields K, H. In a full model of the type proposed by Cummings, the curvature fields are generated from the biochemistry. Before this can be investigated it is necessary to have a means of computational model to simulate the 'growth' of simple organisms. It is this computational model that will be demonstrated here. The investigation of a suitable reaction scheme that is capable of generating asymmetric geometries is the subject of our future work. For the purposes of this demonstration, it is necessary only to generate the curvatures K and H somehow in order to model the shape change. The approach adopted here is then to compute K and H as known components in the shape formation model and then to demonstrate that the resulting model can compute the shape change of organisms.

4.1. Curvature formulation

In order to artificially generate the curvature fields of a surface, the surface has to be first parametrised on known coordinates u_1, u_2. (Note that the u_1, u_2 parametrisation of the curvature fields may or may not coincide with the curvilinear coordinates ξ_1, ξ_2.) Suppose the spherical polar coordinate is used for the parametrisation

$$\mathbf{r}(u_1, u_2) = (R\cos u_1 \cos u_2, R\cos u_1 \sin u_2, R\sin u_1), \tag{26}$$

where $R = R(u_1, u_2)$ is a radius function. The curvatures K, H can be generated using the following formulae (see [22]):

$$K = \frac{\langle \mathbf{r}_{u_1u_1}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle\langle \mathbf{r}_{u_2u_2}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle - \langle \mathbf{r}_{u_1u_2}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle^2}{\left(\|\mathbf{r}_{u_1}\|^2\|\mathbf{r}_{u_2}\|^2 - [\mathbf{r}_{u_1}\cdot\mathbf{r}_{u_2}]^2\right)^2}, \tag{27}$$

$$H = \frac{\langle \mathbf{r}_{u_1u_1}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle\|\mathbf{r}_{u_2}\|^2 - 2\langle \mathbf{r}_{u_1u_2}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle[\mathbf{r}_{u_1}\cdot\mathbf{r}_{u_2}] + \langle \mathbf{r}_{u_2u_2}\mathbf{r}_{u_1}\mathbf{r}_{u_2}\rangle\|\mathbf{r}_{u_1}\|^2}{2\left(\|\mathbf{r}_{u_1}\|^2\|\mathbf{r}_{u_2}\|^2 - [\mathbf{r}_{u_1}\cdot\mathbf{r}_{u_2}]^2\right)^{3/2}}, \tag{28}$$

where the subscripts u_1, u_2 of $\mathbf{r}$ denote their corresponding partial derivatives and the operator $\langle\mathbf{abc}\rangle$ denotes the triple product $\mathbf{a}\cdot\mathbf{b}\times\mathbf{c}$. With these formulae, the remaining task is the construction of the radius function $R(u_1, u_2)$ for the generation of the desired geometry, which, in this case, should exhibit asymmetry.

4.2. An example of asymmetric geometry: tentacle

A simple form of asymmetry can be obtained by superposition of tentacles onto a simple axisymmetric geometry such as a sphere. In terms of spherical polar coordinate (see (26)) the radius function for a sphere is simply a constant R_s. The radius function for a tentacle unit can be generated by the following function:

$$\begin{aligned} R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j) = \frac{fA^j}{4}&\left(1 + \operatorname{erf}[p_1^j(u_1 - p_2^j)]\right)\left(\operatorname{erfc}[p_3^j(u_1 - p_2^j)]\right)\left(1 + \operatorname{erf}[q_1^j(u_2 - q_2^j\right. \\ &\left. - q_3^j)]\right)\left(\operatorname{erfc}[q_1^j(u_2 + q_2^j - q_3^j)]\right), \end{aligned} \tag{29}$$

where A is the length of the fully grown tentacle and $f(t)$ controls the growth of the tentacle as a function of time and the ps and qs are the shape controlling parameters. Parameters p_1^j, p_3^j, q_1^j and q_2^j control the aspect ratio of the tentacle. The larger the numbers, the more elongated the tentacle. The parameter (p_2^j, q_3^j) is the location of the tentacle in (u_1,u_2) coordinate. The superscript j denotes the index of the tentacle unit, thus $R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j)$ stands for radius function of the jth tentacle. Suppose there are n number of tentacles on the surface of a sphere of radius R_s, the radius function R of the complete structure is then generated by the superposition of all the tentacle units onto R_s

$$R(u_1, u_2) = R_s + \sum_{j=1}^{n} R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j). \tag{30}$$

The partial derivatives of $\mathbf{r}$ needed for the computation of K and H ((27) and (28)) can be obtained by the trivial exercise of differentiating (29) and substituting into the spherical polar coordinates derivatives defined by (26).

4.3. Mesh generation and curvatures mapping

In Sections 4.1 and 4.2, the curvatures are generated in spherical polar coordinate (u_1, u_2), they have to be mapped onto the curvilinear coordinate (ξ_1, ξ_2) before they can be used in solving the mesh generation equation. Nevertheless, it has to be realised that this mapping from (u_1, u_2) to (ξ_1, ξ_2) are yet to be found.

Therefore, the mapping of local curvatures to the corresponding location in (ξ_1, ξ_2) is then solved simultaneously with the mesh generation equation, as is shown in the next section.

5. Mapping of curvature fields

The mapping of local curvature value from coordinate (u_1, u_2) to (ξ_1, ξ_2) can be envisaged as a procedure of mesh improvement, i.e., the generation of a mesh (ξ_1, ξ_2) that has the desired qualities, such as the orthogonal and isothermal properties in this case, from (u_1, u_2) that does not. This method has been used to improve the mesh quality of an existing mesh that has been generated using other technique such as NURBS or algebraic grid generators [16,18]. It has also been used in [14] to generate conformal mappings with different mesh aspect ratios on various 2D and 3D domains. Here, it is used to provide the desired geometry on which an isothermal mesh is to be generated. The governing equation can be obtained from the transformation of (13) through the change of variables from (x_1, x_2, x_3) to (u_1, u_2) (see Appendix D). This yields the equation

$$g\left\{\frac{\partial^2}{\partial\xi_1^2}+\frac{\partial^2}{\partial\xi_2^2}\right\}\mathbf{u}-J^2\bar{\Delta}_2\mathbf{u}-\frac{\partial\mathbf{u}}{\partial\sigma}=0. \tag{31}$$

The Laplace–Beltrami operator applied to the components of **u** is given by:

$$\begin{aligned}\bar{\Delta}_2u_1&=\sqrt{\bar{G}_3}\left[\frac{\partial}{\partial u_1}\left(\frac{\bar{g}_{22}}{\sqrt{\bar{G}_3}}\right)-\frac{\partial}{\partial u_2}\left(\frac{\bar{g}_{12}}{\sqrt{\bar{G}_3}}\right)\right],\\ \bar{\Delta}_2u_2&=\sqrt{\bar{G}_3}\left[\frac{\partial}{\partial u_2}\left(\frac{\bar{g}_{11}}{\sqrt{\bar{G}_3}}\right)-\frac{\partial}{\partial u_1}\left(\frac{\bar{g}_{12}}{\sqrt{\bar{G}_3}}\right)\right],\end{aligned} \tag{32}$$

where the variables J, $\bar{g}_{11}$, $\bar{g}_{12}$ and $\bar{g}_{22}$ are given by

$$J=\frac{\partial u_1}{\partial\xi_1}\frac{\partial u_2}{\partial\xi_2}-\frac{\partial u_1}{\partial\xi_2}\frac{\partial u_2}{\partial\xi_1},\quad \bar{g}_{11}\equiv\frac{\partial\mathbf{r}}{\partial u_1}\cdot\frac{\partial\mathbf{r}}{\partial u_1},\quad \bar{g}_{12}\equiv\frac{\partial\mathbf{r}}{\partial u_1}\cdot\frac{\partial\mathbf{r}}{\partial u_2},\quad \bar{g}_{22}\equiv\frac{\partial\mathbf{r}}{\partial u_2}\cdot\frac{\partial\mathbf{r}}{\partial u_2}. \tag{33}$$

Since the geometry $\mathbf{r}(x_1, x_2, x_3)$ are parameterised in terms of (u_1, u_2), the quantities $\bar{g}_{ij}$ $(i, j = 1, 2)$ are exact.

Eq. (31) is solved with the following boundary conditions: at $\xi_1 = -l$, sphere solution is used

$$u_1 = \sin^{-1}\tanh\xi_1, \quad u_2 = \xi_2 \tag{34}$$

at $\xi_1 = l$, the orthogonal property is enforced, i.e., $g_{12} = 0$ is incorporated, this gives

$$\bar{g}_{12}\frac{\partial u_1}{\partial\xi_2}+\bar{g}_{22}\frac{\partial u_2}{\partial\xi_2}=0. \tag{35}$$

This is used in conjunction with the following condition:

$$\frac{\partial u_1}{\partial\xi_1}=\sqrt{1-\tanh^2\xi_1}, \tag{36}$$

which has taken into account the assumption that the geometry approaches to a sphere in the vicinity of the region $\xi_1 = l$. Along the boundaries $\xi_2 = 0, 2\pi$, periodic conditions are incorporated except for the functional value of u_2 which takes the form,

$$u_2(\xi_2 = 0) = u_2(\xi_2 = 2\pi) - 2\pi. \tag{37}$$

These boundary conditions are applied in the same way as those described in Section 3.3.

6. Numerical implementation (Vlugr2)

Vlugr2 (Vectorised Local Uniform Grid Refinement 2D) [2] is used to solve the system of differential equations, (20), and (Eq. (31)). This code has been designed to solve initial boundary value problems for general systems of time-dependent partial differential equations in two dimensions of the form

$$\mathscr{F}\left(t, \xi_1, \xi_2, \mathbf{r}, \frac{\partial \mathbf{r}}{\partial t}, \frac{\partial \mathbf{r}}{\partial \xi_1}, \frac{\partial \mathbf{r}}{\partial \xi_2}, \frac{\partial^2 \mathbf{r}}{\partial \xi_1^2}, \frac{\partial^2 \mathbf{r}}{\partial \xi_1 \, \partial \xi_2}, \frac{\partial^2 \mathbf{r}}{\partial \xi_2^2}\right) = 0, \quad (\xi_1, \xi_2) \in \Omega, \ \ t > t_0, \tag{38}$$

where $\mathbf{r}$ is the solution vector and Ω denotes an arbitrary domain that can be described by right-angled polygons. Along the boundary $\partial\Omega$, the boundary condition is assumed to have the form:

$$\mathscr{B}\left(t, \xi_1, \xi_2, \mathbf{r}, \frac{\partial \mathbf{r}}{\partial \xi_1}, \frac{\partial \mathbf{r}}{\partial \xi_2}\right) = 0, \quad (\xi_1, \xi_2) \in \partial\Omega, \ \ t > t_0, \tag{39}$$

and the initial condition is assumed to be

$$\mathbf{r}(t_0, \xi_1, \xi_2) = \mathbf{r}_0(\xi_1, \xi_2), \quad (\xi_1, \xi_2) \in \Omega \cup \partial\Omega. \tag{40}$$

The system of PDEs are solved using the *method of lines* [28]. The underlying principle is that spatial discretisation is used to transform the PDEs in both space and time into a system of ODEs in time alone. The spatial discretisation can be done using various schemes such as finite differences or finite elements with desired order of accuracy while the resultant system of ODEs can then be solved using standard methods such as Runge–Kutta or BDF to preserve the accuracy and stability in the time integration. Modified Newton's method in combination with sparse matrix solver GMRES is used to solve the resultant system of nonlinear algebraic equations (see [2]).

Finite differences are used in Vlugr2 for spatial discretisation with central difference formulae to approximate both the first and the second derivatives

$$\frac{\partial \mathbf{r}}{\partial \xi_i} = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2\Delta\xi_i}, \quad \frac{\partial^2 \mathbf{r}}{\partial \xi_i^2} = \frac{\mathbf{r}_{i+1} - \mathbf{r}_i + \mathbf{r}_{i-1}}{\Delta\xi_i^2}, \tag{41}$$

which are both second order. Along the boundary, the derivatives are approximated using one-sided formula

$$\frac{\partial \mathbf{r}}{\partial \xi_i} = \frac{3\mathbf{r}_n - 4\mathbf{r}_{n-1} + \mathbf{r}_{n-2}}{2\xi_i}. \tag{42}$$

Spatial mesh refinement is used to improve solution accuracy with a user-defined maximum mesh level. The solution accuracy is monitored by the curvature function `spcmon`

$$\texttt{spcmon}(i,j) \propto \left\| \left\{ \left| \Delta\xi_1^2 \frac{\partial^2 r_c}{\partial \xi_1^2} \right| + \left| \Delta\xi_2^2 \frac{\partial^2 r_c}{\partial \xi_2^2} \right| \right\} \right\|_\infty, \tag{43}$$

where i, j denote the current grid point, $\Delta\xi_1$ and $\Delta\xi_2$ are the grid size of the current mesh level, r_c denotes the component c of the solution vector $\mathbf{r}$ and the norm is the maximum among all solution components at the node (i,j). When the local value of spcmon exceeds its tolerance, the current grid point and each of its eight immediate neighbourhood will be subdivided uniformly into four sub-grid points. Thus, patches of refined mesh are formed in the next mesh level where it is required. The system of PDEs are then recomputed on this refined level with the initial estimate taken from the solution of the previous mesh level. Interpolations are performed on those newly created grid points. Refinement terminates only when either the solution accuracy has been reached or the maximum level has been reached.

The time integration in Vlugr2 is performed using second order BDF (Backward Differentiation Formula) method with variable time step control in order to take into account the stiffness of the time rate of change of solution. The formula for the time derivative approximation is given by

$$\frac{\partial \mathbf{r}}{\partial t} = a_0\mathbf{r}^{n+1} + a_1\mathbf{r}^n + a_2\mathbf{r}^{n-1}, \tag{44}$$

where

$$a_0 = \frac{1+2\alpha}{1+\alpha}\frac{1}{\Delta t}, \quad a_1 = \frac{(1+\alpha)^2}{1+\alpha}\frac{1}{\Delta t}, \quad a_2 = \frac{\alpha^2}{1+\alpha}\frac{1}{\Delta t}, \quad \alpha = \frac{\Delta t}{\Delta t_{\mathrm{old}}}. \tag{45}$$

Backward Euler ($\alpha = 0$) is used at the first step. The time step Δt is monitored at a all mesh levels and is modified according to the function $\mathtt{timmon}(L)$ which depends on the time rate of change of solution components at mesh level L

$$\mathtt{timmon}(L) \propto \left\| \Delta t \left\{ \left| \frac{\partial r_c}{\partial t} \right| \right\} \right\|_\infty, \quad \Delta t_{\mathrm{new}} = \frac{1}{2\mathtt{timmon}_{\max}} \Delta t, \tag{46}$$

where Δt_{new} denotes the new time step and $\mathtt{timmon}_{\max}$ is the maximum value of timmon among all mesh levels.

When all the spatial and temporal accuracy has been satisfied, computation advances to the next time step. The initial estimate of the next step is injected from the solution of the highest mesh level at the previous step.

6.1. Computational mesh refinement strategy

A user defined mesh refinement facility is also provided in the subroutine chspcm in Vlugr2 so that particular regions can be refined regardless of the local curvature of the solutions. This additional facility is incorporated to monitor the local Gauss curvature $K_{i,j}$ to insert extra refinement region where either of the following occurs:

- $K_{i+1,j} \times K_{i,j} < 0$, this is incorporated in order to monitor region where invagination or evagination takes place;
- $\partial K/\partial\xi_1|_{i+1,j} \times \partial K/\partial\xi_1|_{i,j} < 0$, this is incorporated simply to detect any large rate of change of local K.

These two monitor criteria are incorporated so that both changes in sign and large spatial changes in K can be resolved as they incur the formation of 'sharp' edges and saddle regions, thus represent crucial geometrical development and have to be resolved in order to capture the corresponding geometry changes.

6.2. Overview of computational procedures

The overall computational procedures of mesh generation algorithm can now be stated (see Fig. 5). At the beginning of a time step:

1. An initial estimate of the solutions of $\mathbf{u}^0$ and $\mathbf{r}^0$ are taken from the solution of the previous time step.
2. Vlugr2 is then called to form the residual of (31) and (20), at every point on the computational mesh:

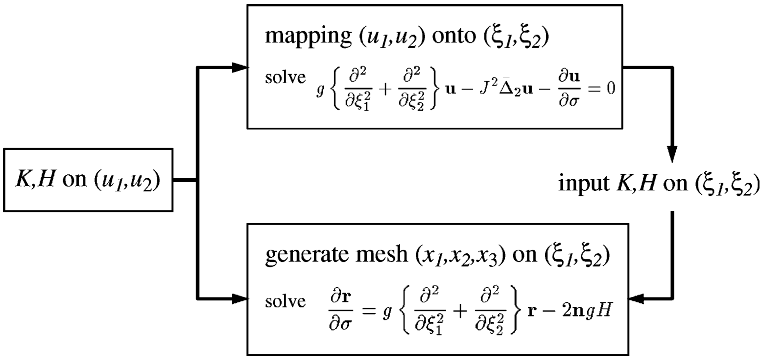
mapping (u_1,u_2) onto (ξ_1,ξ_2)
solve $g\left\{\frac{\partial^2}{\partial\xi_1^2}+\frac{\partial^2}{\partial\xi_2^2}\right\}\mathbf{u}-J^2\bar{\Delta}_2\mathbf{u}-\frac{\partial\mathbf{u}}{\partial\sigma}=0$
K,H on (u_1,u_2)
input K,H on (ξ_1,ξ_2)
generate mesh (x_1,x_2,x_3) on (ξ_1,ξ_2)
solve $\frac{\partial\mathbf{r}}{\partial\sigma}=g\left\{\frac{\partial^2}{\partial\xi_1^2}+\frac{\partial^2}{\partial\xi_2^2}\right\}\mathbf{r}-2\mathbf{n}gH$

Fig. 5. An overview of computational scheme for mesh generation with artificial input of K,H.

(a) use $\mathbf{u}^0$ to compute $\bar{g}_{11}$, $\bar{g}_{12}$ and $\bar{g}_{22}$ from geometry specifications; $\bar{\Delta}_2u_1$ and $\bar{\Delta}_2u_2$ using (32);

(b) compute the g by using the formula

$$g=\bar{g}_{11}\frac{\partial u_1}{\partial\xi_1}^2+\bar{g}_{12}\frac{\partial u_1}{\partial\xi_1}\frac{\partial u_2}{\partial\xi_1}+\bar{g}_{22}\frac{\partial u_2}{\partial\xi_1}^2;$$

(c) substitute g, J, $\bar{\Delta}_2u_1$, $\bar{\Delta}_2u_2$ into (31) to form the associated residual;

(d) input $\mathbf{u}^0$ into (28) in combination with geometry specification to compute the exact H;

(e) use $\mathbf{r}^0$ to compute g using

$$g=\left(\frac{\partial x_1}{\partial\xi_1}\right)^2+\left(\frac{\partial x_2}{\partial\xi_1}\right)^2+\left(\frac{\partial x_3}{\partial\xi_1}\right)^2,$$

and $\mathbf{n}$ using (14);

(f) substitute g, $\mathbf{n}$ and H into (20) to obtain its residual.

3. The system of residual is then iterated using Newton's method to give $\mathbf{u}^1$ and $\mathbf{r}^1$,
4. The solution accuracy is measured, the complete procedure is repeated until subsequence modification in solution falls below a tolerance.

The computation then advances to next time step. Note that the above procedure has not included the solution algorithm in Vlugr2, it merely outlines the steps involved in forming the residual associated with the governing equations.

7. Demonstrations of growing tentacles

Three examples are illustrated: (1) a single tentacle on a sphere, this shows the simplest non-axisymmetric geometry that can be modelled, the resultant geometry however is bilaterally symmetrical; (2) three unevenly spaced tentacles on a sphere, this shows a geometry that is neither axisymmetrical nor bilaterally symmetrical; (3) unevenly spaced multi-tentacles of different sizes and shapes on a sphere, this demonstrates a general asymmetric geometry, six tentacles were used in the experiment.

In all examples, the tentacle structure is generated using (30). The parameters defining the tentacles for all examples are summarised in Table 1. The parameters A, p_1 to p_3 and q_1 to q_3 are defined in (30). In Example 1, the radius function is given by

Table 1
Parameter values used for tentacle specification

jth Tentacle unit	A^j	p_1^j	p_2^j	p_3^j	q_1^j	q_2^j	q_3^j
1	2	2	0.3π	8	3	0.3	π
2	2	2	0.2π	8	3	0.3	$0.\dot{4}\pi$
3	2	2	0.2π	8	4	0.2	$1.\dot{4}\pi$
4	1	4	-0.1π	6	5	0.3	$1.0\dot{5}\pi$
5	1.2	2	0.0	8	5	0.3	$1.\dot{3}\pi$
6	1.5	2	-0.1π	7	4	0.3	$1.\dot{6}\pi$

$$R(u_1, u_2) = R_s + R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j), \quad j = 1, \tag{47}$$

where $j = 1$ denotes the first tentacle parameters set is used (see Table 1). Similarly, in Example 2, the overall radius function $R(u_1, u_2)$ of the three tentacle structure is obtained by the summation of three individual tentacle units

$$R(u_1, u_2) = R_s + \sum_{j=1}^{3} R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j), \tag{48}$$

where the first three parameters sets are used. In Example 3, the overall radius function $R(u_1, u_2)$ is formed by the summation of all six tentacle sets

$$R(u_1, u_2) = R_s + \sum_{j=1}^{6} R_t^j(u_1, u_2, A^j, p_1^j, p_2^j, p_3^j, q_1^j, q_2^j, q_3^j). \tag{49}$$

The Euclidean geometry of the three fully grown tentacle organisms are illustrated in Figs. 6(a)–(c). The computational meshes corresponded to the three fully grown tentacle organisms are illustrated in Figs. 6(d)–(f). Only the region $[-2 < \xi_1 < 2] \times [0 < \xi_2 < 2\pi]$ is drawn since no refinement was performed on the remaining regions. It can be seen that the mesh was refined in one region only in the case of a single tentacle organism, whereas the mesh was refined in three different regions in the case of the three tentacle organism, which corresponds to the three tentacles, and in the case of the six tentacle organism, refinement were performed in six different regions.

Notice in Figs. 6(d)–(f), the location of the tentacle units have been mapped from the spherical coordinate u_1, u_2 to the surface coordinate ξ_1, ξ_2. For example, in Example 1, the location of the single tentacle is located at $(u_1, u_2) = (p_2^1, q_3^1) = (0.3\pi, \pi)$, after the mapping, the corresponding location on the surface is approximately $(\xi_1, \xi_2) = (0.9, \pi)$. Thus, the mapping has been distorted in the vicinity of the tentacle, and is deviated from that of a sphere. The mapping of a sphere is given by the Mercator transformation (see Appendix C)

$$\sin u_1 = \tanh \xi_1. \tag{50}$$

Upon the substitution of $u_1 = 0.3\pi$ into (50) yields $\xi_1 = 1.1$. In contrast, the u_2 coordinate remains close to the ξ_2. This is because of the bilateral symmetric nature of the tentacle. The location of the six tentacles in u_1, u_2 coordinates and in the corresponding ξ_1, ξ_2 coordinates are listed in Table 2. The corresponding location in ξ_1 mapped by using the Mercator transformation are also included for comparison. (Note that the ξ_2 is not affected in the Mercator transformation and is equal to u_2.) It is noticed that the distortion of the mesh due to the formation of tentacle units is localised around the tentacles. However, when two tentacles are sufficiently close, the corresponding meshes can affect each other. This can be observed in the

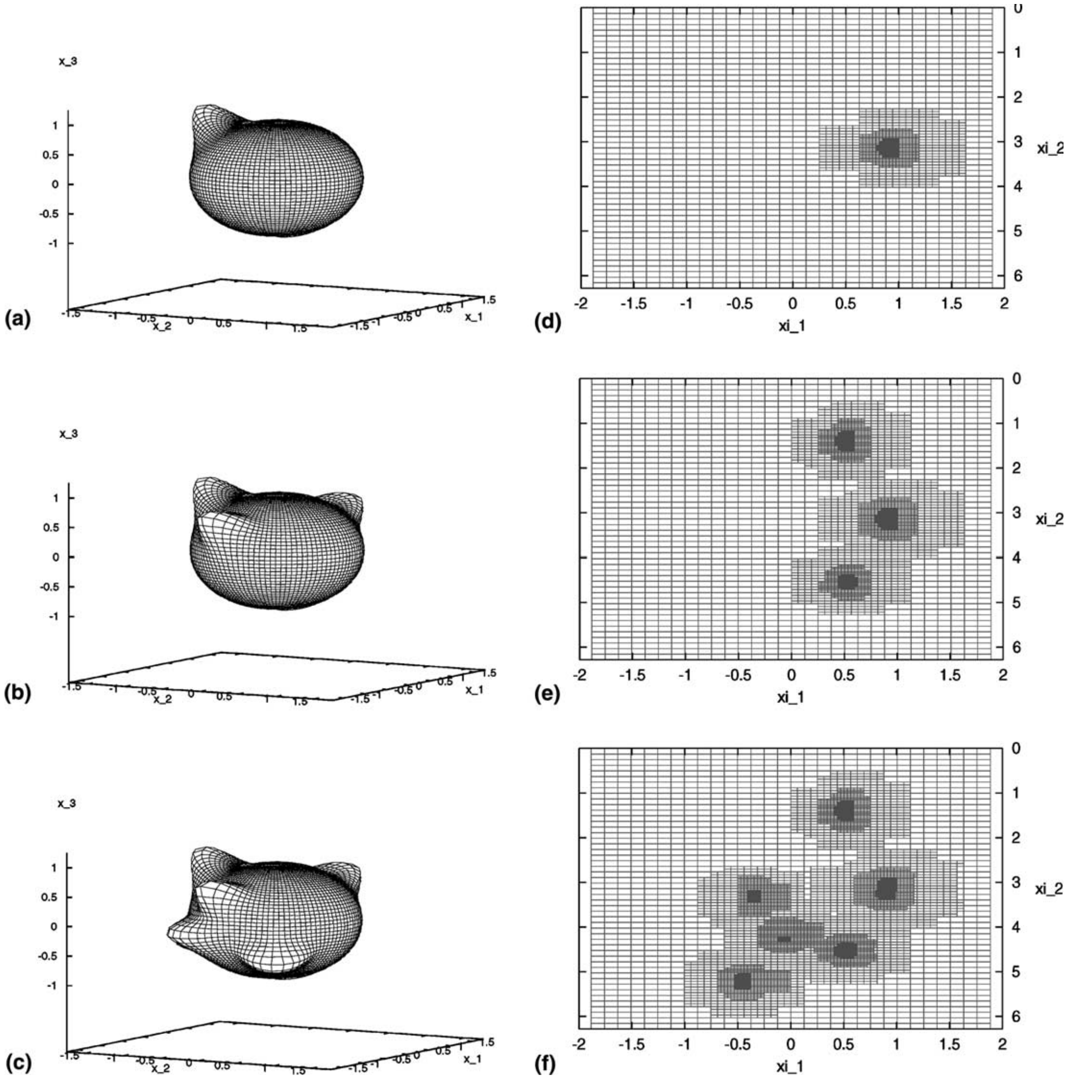

Fig. 6. The Euclidean geometry of (a) a single tentacle unit, (b) three tentacle units and (c) six tentacle units growing on a sphere; and the associated computational meshes of (d) the single tentacle, (e) the three tentacles and (f) the six tentacles.

Table 2
A comparison of the location of the tentacle units in u_1, u_2 coordinates and in the corresponding ξ_1, ξ_2 coordinates. The coordinate ξ_1^m is the location assuming the Mercator transformation

*j*th Tentacle units	u_1	u_2	ξ_1	ξ_2	ξ_1^m
1	0.3π	π	0.90	π	1.1
2	0.2π	$0.\dot{4}\pi$	0.50	0.44π	0.67
3	0.2π	$1.\dot{4}\pi$	0.50	1.43π	0.67
4	-0.1π	$1.0\dot{5}\pi$	−0.38	1.03π	−0.32
5	0.0	$1.\dot{3}\pi$	0.125	1.35π	0.0
6	-0.1π	$1.\dot{6}\pi$	−0.44	1.67π	−0.32

case of the 4th, 5th and the 6th tentacles. Note that even though their u_1 coordinates are identical, the corresponding ξ_1 coordinates are not the same.

The metric tensor g and the corresponding Gauss curvature K of the three examples of tentacle organisms are illustrated in Fig. 7. This figure shows that when the Gauss curvature K of a tentacle unit takes the form of a 'spike' with a region of negative values surrounding it. In the case of the example with three tentacles, it can be seen that there are three spikes surrounded by a negative region. Similarly, there are six spikes in Example 3. These examples therefore suggest that the Gauss curvature of an organism with tentacles can be generated by the superposition of such K (a spike with a surrounding negative region) units onto an existing K distribution. This knowledge is particular useful when a complex organism is generated using only the Gauss curvature. This also provides a feasible method towards the engineering of the parametrisation of the curvatures using morphogens' concentration.

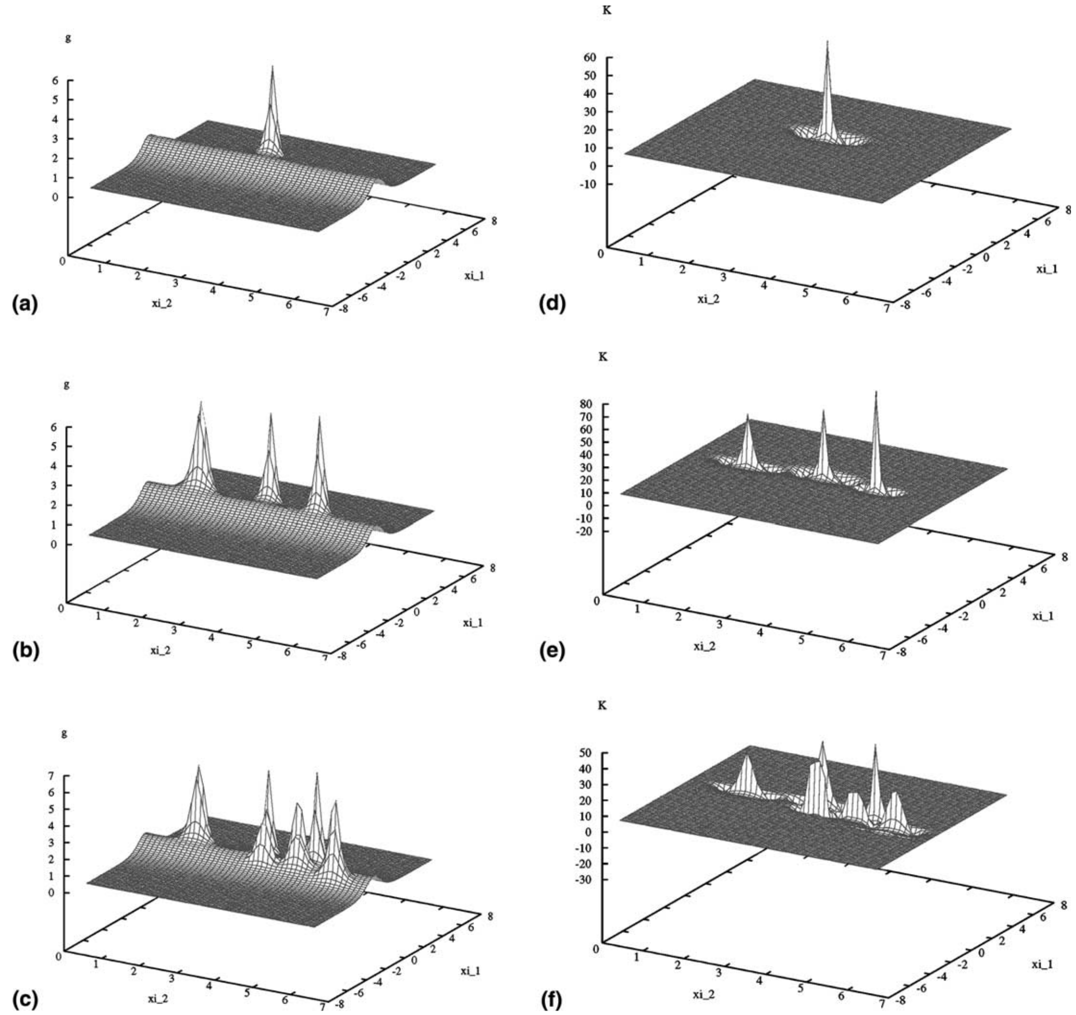

Fig. 7. The metric tensor g of (a) a single tentacle unit, (b) three tentacle units and (c) six tentacle units growing on a sphere; and the associated Gauss curvature of (d) the single tentacle, (e) the three tentacles and (f) the six tentacles.

8. Towards generating realistic organisms

Part of the more extended goal of this work is to simulate the morphological evolution of a wide range of organisms such as hydroid polyps, medusae and hydranth (see Fig. 8). As have been demonstrated in the generation of tentacle organisms, the organisms can be created by the superposition of various components of the final geometry onto the initial sphere. For example, the geometry of the polyp can be dissected into the components of: (a) evagination (protruding or elongation in the formation of limb bud); (b) invagination (the in-folding of a region of cells, like the indenting of a soft rubber ball when poked, important of the formation of a mouth and through-gut during development); and (c) a ring of tentacles (for food capturing in certain organisms). In terms of spherical polar coordinate, both the invagination and evagination can be generated using the same parametrisation

$$R_{vg}(u_1) = \frac{f_{vg}A}{2}[1 + \cos(\pi - \phi)]\tanh(c\phi^d), \quad \phi = \frac{\pi(u_1 - b)}{\pi/2 - b}, \tag{51}$$

where A is positive in the case of evagination and negative in the case of invagination, and f is a function of time. Instead of superposing individual tentacles to form a ring, the ring of tentacles is generated as a completed unit using the formula

$$R_{\mathrm{ring}}(u_1, u_2) = f_{\mathrm{ring}}A\left\{\frac{1 - \cos[n(u_2 - c)]}{2}\right\}^m\left\{\frac{1 + \cos[(u_1 - b)/p]}{2}\right\}. \tag{52}$$

The entire organism is therefore generated by the summation of all the radius functions

$$R(u_1, u_2) = R_{\mathrm{s}} + R_{vg}(u_1) + R_{\mathrm{ring}}(u_1, u_2), \tag{53}$$

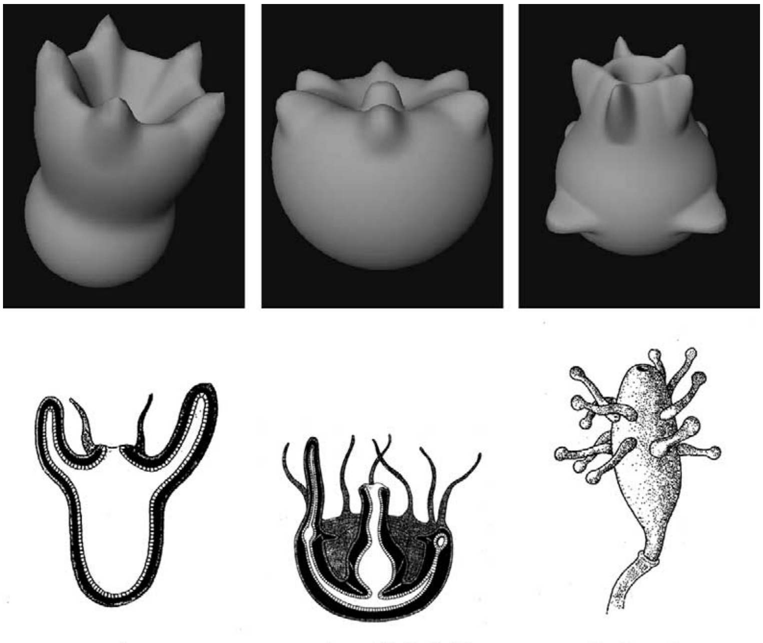

Fig. 8. A set of computationally simulated cnidarians (top row) and the corresponding graphical illustrations (bottom row).

where R_s is the radius of the initial sphere as before. The function $f(t)$ takes the form

$$f(t) = \begin{cases} (t - t_0)/(t_1 - t_0) & \text{for } t_0 \leqslant t \leqslant t_1, \\ 1 & \text{for } t \geqslant t_1, \end{cases} \tag{54}$$

where the interval $[t_0, t_1] = [0, 0.5]$ determines the evolution of both the invagination and evagination while the interval $[t_0, t_1] = [0.5, 1.0]$ determines the evolution of the tentacle ring. This allow different components to evolve one after another. The parameters used in the case of the polyp (shown in Fig. 8) are listed in Table 3. These parametrisations and their derivatives are then used for the computations of the Gauss and the Mean curvatures as in the case of the tentacle examples.

The evolution of the simulated polyp is illustrated in Fig. 9. Parts (a)–(d) of this figure show the sphere evaginates on the top into a bud while it invaginates simultaneous at the centre the bud to form a 'mouth'. From parts (e)–(h) onward, both the evagination and invagination have already terminated, while the ring of tentacles start to grow. At part (h), the completed organism is fully grown. The Mean curvature of the corresponding snapshots are shown in Fig. 10.

Table 3
The parameter values used for polyp specification

Parameters	Evagination	Invagination	Tentacle ring
A	1.4	−1.2	0.4
b	$0.\dot{5}\pi$	$0.\dot{3}\pi$	$\pi/2 - 2p$
c	2	4	0
d	4	4	/
p	/	/	0.06π
n	/	/	5
m	/	/	4

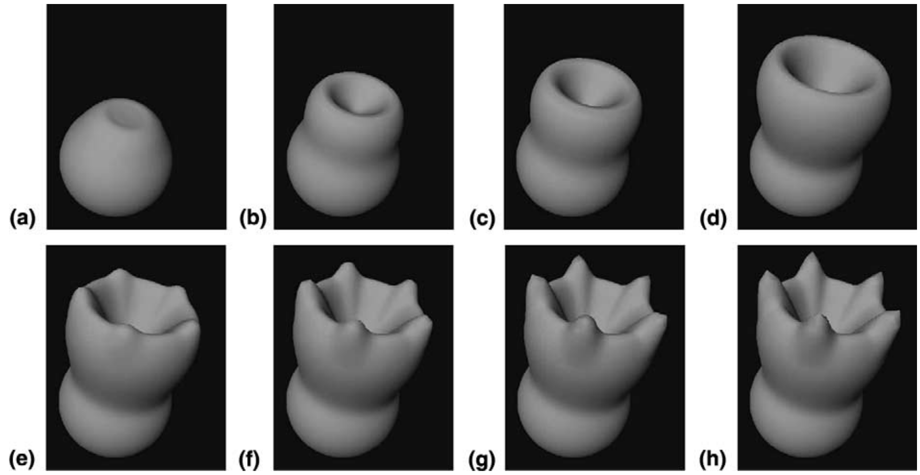

Fig. 9. The snapshots of the evolution of a sphere into a polyp. The frames represent the corresponding stages of growth at approximately: (a) 0.12; (b) 0.24; (c) 0.36; (d) 0.48; (e) 0.6; (f) 0.72; (g) 0.84; and (h) 0.96 time unit, respectively.

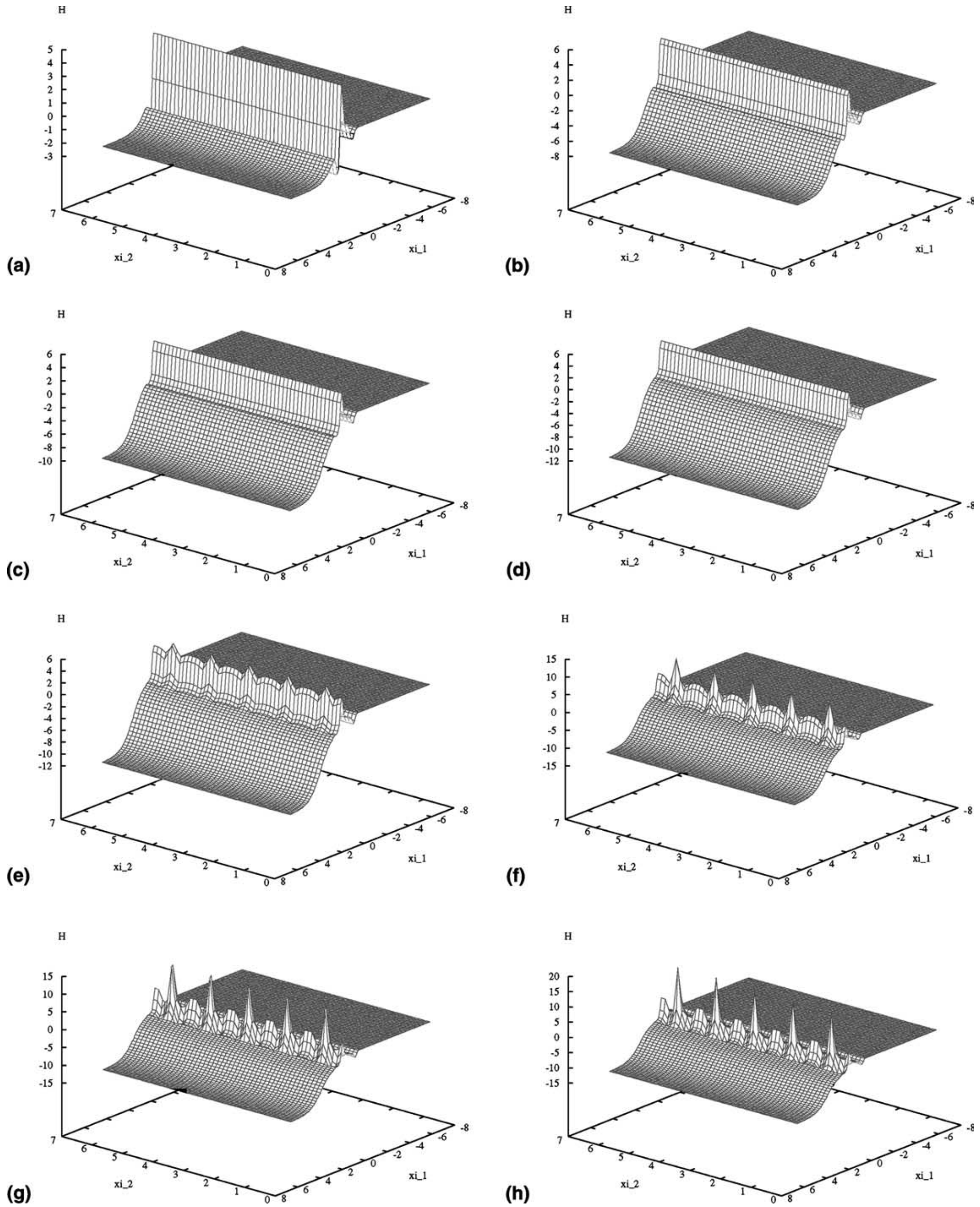

Fig. 10. The snapshots of the evolution of the Mean curvature associated with the polyp. The frames represent the corresponding stages of growth at approximately: (a) 0.12; (b) 0.24; (c) 0.36; (d) 0.48; (e) 0.6; (f) 0.72; (g) 0.84; and (h) 0.96 time unit, respectively.

9. Conclusion

The method defined above and the results shown demonstrate the ability to model the shape change of simple organisms. In particular the idea of Cummings that generated Mean and Gauss curvatures may be used to compute non-axisymmetric changing surface geometries has been shown to work in a practical system for generating shape change. It is now possible to investigate the detailed chemical interactions that may be plausible agents for shape change. Examples of such interactions are given by Cummings in recent work [10,11]. Research into which agents to use will form the basis of our future work.

Acknowledgements

The author thanks F.W. Cummings, B.C. Goodwin and D.C. Hogg for the numerous discussions, and their active collaboration on this work; and EPSRC for the financial support.

Appendix A. A brief description of surface curvatures

The nature of a surface, i.e., whether it is concave or convex, around a general point, p, can be described by the rate of change of tangents (or the curvatures) at that point, just as a curve can be described by same quantity. Although it is possible to draw infinitely many curves passing through p on a surface. There are two curves that intersect in directions perpendicular with each other and are associated with the maximum and the minimum curvatures at p. The two curvatures, denoted by $k_{\rm I}$ and $k_{\rm II}$, are called the principal curvatures (illustrated in Fig. 11). It is a convention to associate a positive k with a curve on a convex surface and a negative k on a concave one. Therefore, when the curvature k varies between two positive values, i.e., when both $k_{\rm I}$ and $k_{\rm II}$ are positive, they describe a convex surface. However, when $k_{\rm I}$ and $k_{\rm II}$ span from negative to positive, they describe a saddle surface. When $k_{\rm I}$ and $k_{\rm II}$ span from zero to a non-zero value, they describe either a ridge (denoted by a positive maximum and zero minimum curvature) or a valley (characterised by a zero maximum and a negative minimum curvature). When both $k_{\rm I}$ and $k_{\rm II}$ are zero, the surface is a plane. It is possible to describe a surface with the Gauss and the Mean curvatures K and H which are defined in terms of $k_{\rm I}$ and $k_{\rm II}$ as follows,

$$K = k_{\rm I}k_{\rm II}, \quad H = (k_{\rm I} + k_{\rm II})/2. \tag{A.1}$$

A brief graphical descriptions of the nature of a surface and its corresponding K and H are illustrated in Fig. 12.

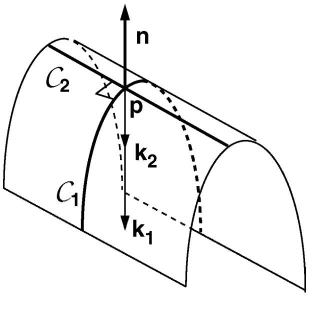

Fig. 11. An illustration of the curves C_1 and C_2, passing through a general point p on a surface, that are associated with the maximum and the minimum curvatures $k_{\rm I}$ and $k_{\rm II}$. The vector **n** denotes the surface normal.

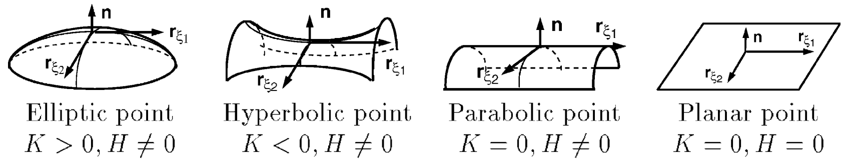

Fig. 12. A graphical description of various natures of a surface and the associated K and H. The surface tangents $\mathbf{r}_{\xi_1}$, $\mathbf{r}_{\xi_2}$ and $\mathbf{n}$ are also shown.

Appendix B. Derivation of the mesh generation equation

The mesh generator can be derived from the equations of Gauss (see [33–35,37])

$$\frac{\partial^2 \mathbf{r}}{\partial \xi_1 \, \partial \xi_1} = \Upsilon^1_{11} \frac{\partial \mathbf{r}}{\partial \xi_1} + \Upsilon^2_{11} \frac{\partial \mathbf{r}}{\partial \xi_2} + b_{11} \frac{\partial \mathbf{r}}{\partial \xi_3}, \tag{B.1}$$

$$\frac{\partial^2 \mathbf{r}}{\partial \xi_1 \, \partial \xi_2} = \Upsilon^1_{12} \frac{\partial \mathbf{r}}{\partial \xi_1} + \Upsilon^2_{12} \frac{\partial \mathbf{r}}{\partial \xi_2} + b_{12} \frac{\partial \mathbf{r}}{\partial \xi_3}, \tag{B.2}$$

$$\frac{\partial^2 \mathbf{r}}{\partial \xi_2 \, \partial \xi_2} = \Upsilon^1_{22} \frac{\partial \mathbf{r}}{\partial \xi_1} + \Upsilon^2_{22} \frac{\partial \mathbf{r}}{\partial \xi_2} + b_{22} \frac{\partial \mathbf{r}}{\partial \xi_3}, \tag{B.3}$$

which are the rate of change of the two tangent vectors $\partial \mathbf{r}/\partial \xi_1, \partial \mathbf{r}/\partial \xi_2$ in the two directions ξ_1, ξ_2, i.e., their curvatures. Performing $g_{22}\times$ (B.1) $-2g_{12}\times$ (B.2) $+g_{11}\times$ (B.3) yields

$$\begin{aligned} g_{22} \frac{\partial^2 \mathbf{r}}{\partial \xi_1^2} - 2g_{12} \frac{\partial^2 \mathbf{r}}{\partial \xi_1 \, \partial \xi_2} + g_{11} \frac{\partial^2 \mathbf{r}}{\partial \xi_2^2} &= \{g_{22}\Upsilon^1_{11} - 2g_{12}\Upsilon^1_{12} + g_{11}\Upsilon^1_{22}\} \frac{\partial \mathbf{r}}{\partial \xi_1} + \{g_{22}\Upsilon^2_{11} - 2g_{12}\Upsilon^2_{12} \\ &\quad + g_{11}\Upsilon^2_{22}\} \frac{\partial \mathbf{r}}{\partial \xi_2} + \{g_{22}b_{11} - 2g_{12}b_{12} + g_{11}b_{22}\} \frac{\partial \mathbf{r}}{\partial \xi_3}. \end{aligned} \tag{B.4}$$

Using the formulae

$$g^{11} = \frac{g_{22}}{G_3}, \quad g^{12} = g^{21} = -\frac{g_{12}}{G_3}, \quad g^{22} = \frac{g_{11}}{G_3}, \quad G_3 = g_{11}g_{22} - g_{12}^2,$$

$$\Upsilon^n_{ij} = g^{kn}[ij,k], \quad [ij,k] = \frac{1}{2}\left(\frac{\partial g_{ik}}{\partial \xi_j} + \frac{\partial g_{jk}}{\partial \xi_i} - \frac{\partial g_{ij}}{\partial \xi_k} \right), \tag{B.5}$$

and expanding

$$\Upsilon^1_{11} = g^{11}[11,1] + g^{12}[11,2] = \frac{g_{22}}{2G_3} \frac{\partial g_{11}}{\partial \xi_1} - \frac{g_{12}}{2G_3} \left(2\frac{\partial g_{12}}{\partial \xi_1} - \frac{\partial g_{11}}{\partial \xi_2} \right), \tag{B.6}$$

$$\Upsilon^1_{12} = g^{11}[12,1] + g^{12}[12,2] = \frac{g_{22}}{2G_3} \frac{\partial g_{11}}{\partial \xi_2} - \frac{g_{12}}{2G_3} \frac{\partial g_{22}}{\partial \xi_1}, \tag{B.7}$$

$$\Upsilon^1_{22} = g^{11}[22,1] + g^{12}[22,2] = \frac{g_{22}}{2G_3} \left(2\frac{\partial g_{12}}{\partial \xi_2} - \frac{\partial g_{22}}{\partial \xi_1} \right) - \frac{g_{12}}{2G_3} \frac{\partial g_{22}}{\partial \xi_2} \tag{B.8}$$

it can be readily shown that

$$g_{22}\Upsilon^1_{11} - 2g_{12}\Upsilon^1_{12} + g_{11}\Upsilon^1_{22} = -G_3\Delta_2\xi_1 \tag{B.9}$$

and similarly

$$g_{22}\Upsilon^2_{11} - 2g_{12}\Upsilon^2_{12} + g_{11}\Upsilon^2_{22} = -G_3\Delta_2\xi_2 \tag{B.10}$$

where Δ_2 is the Laplace–Beltrami operator,

$$\Delta_2 = \frac{\partial}{\partial\xi_1}\left(\frac{g_{22}}{\sqrt{G_3}}\frac{\partial}{\partial\xi_1} - \frac{g_{12}}{\sqrt{G_3}}\frac{\partial}{\partial\xi_2}\right) + \frac{\partial}{\partial\xi_2}\left(\frac{g_{11}}{\sqrt{G_3}}\frac{\partial}{\partial\xi_2} - \frac{g_{12}}{\sqrt{G_3}}\frac{\partial}{\partial\xi_1}\right). \tag{B.11}$$

Realising that the two principal curvatures k_{I} and k_{II} can be found by finding the extrema from the definition [22]

$$k_n = \frac{b_{11} + 2b_{12}\lambda + b_{22}\lambda^2}{g_{11} + 2g_{12}\lambda + g_{22}\lambda^2}. \tag{B.12}$$

(Here λ is a variable and it denotes the principal direction.) The sum of the two principal curvatures $k_{\mathrm{I}} + k_{\mathrm{II}}$ can be readily obtained as

$$G_3(k_{\mathrm{I}} + k_{\mathrm{II}}) = g_{22}b_{11} - 2g_{12}b_{12} + g_{11}b_{22} = G_3 2H. \tag{B.13}$$

Combining (B.4) and (B.13) and upon realising $\mathbf{n} = \partial\mathbf{r}/\partial\xi_3$, it yields

$$g_{22}\frac{\partial^2\mathbf{r}}{\partial\xi_1^2} - 2g_{12}\frac{\partial^2\mathbf{r}}{\partial\xi_1\,\partial\xi_2} + g_{11}\frac{\partial^2\mathbf{r}}{\partial\xi_2^2} + G_3\left[\Delta_2\xi_1\frac{\partial\mathbf{r}}{\partial\xi_1} + \Delta_2\xi_2\frac{\partial\mathbf{r}}{\partial\xi_2}\right] = \mathbf{n}G_3 2H. \tag{B.14}$$

Appendix C. Mercator transformation

The Mercator transformation relates the spherical polar coordinate $\mathbf{r} = (R\cos\theta\cos\phi, R\cos\theta \sin\phi, R\sin\theta)$ to the orthogonal isothermal curvilinear coordinate $\mathbf{r}(R, \xi_1, \xi_2)$ and can be readily obtained as follows: Upon realising that a sphere is a surface of revolution and that $\xi_2 = \phi$, the curvilinear coordinate can be given by

$$\mathbf{r}(\xi_1, \xi_2) = (F(\xi_1)\cos\xi_2, F(\xi_1)\sin\xi_2, G(\xi_1)). \tag{C.1}$$

By applying the isothermal conditions

$$\frac{\partial\mathbf{r}}{\partial\xi_1}\cdot\frac{\partial\mathbf{r}}{\partial\xi_1} = \left(\frac{\mathrm{d}F}{\mathrm{d}\xi_1}\right)^2 + \left(\frac{\mathrm{d}G}{\mathrm{d}\xi_1}\right)^2 = \frac{\partial\mathbf{r}}{\partial\xi_2}\cdot\frac{\partial\mathbf{r}}{\partial\xi_2} = F(\xi_1)^2. \tag{C.2}$$

Moreover, a comparison with the spherical polar coordinate yields

$$F(\xi_1) = R\cos\theta, \quad G(\xi_1) = R\sin\theta, \tag{C.3}$$

which gives

$$G(\xi_1)^2 = R^2 - F(\xi_1)^2, \quad \text{hence} \quad \left(\frac{\mathrm{d}G}{\mathrm{d}\xi_1}\right)^2 = \frac{F(\xi_1)^2}{R^2 - F(\xi_1)^2}\left(\frac{\mathrm{d}F}{\mathrm{d}\xi_1}\right)^2. \tag{C.4}$$

Upon substitution of (C.4) into (C.2) yields an ode for $F(\xi_1)$

$$\frac{\mathrm{d}F}{\mathrm{d}\xi_1} = \pm\frac{1}{R}F(\xi_1)\sqrt{R^2 - F(\xi_1)^2}. \tag{C.5}$$

(The positive sign is chosen such that the top, $\theta = \pi/2$, corresponds to $\xi_1 = l$.) Upon solving for $F(\xi_1)$ yields

$$F(\xi_1) = \cos\theta = \operatorname{sech}\xi_1, \qquad G(\xi_1) = \sin\theta = \tanh\xi_1, \tag{C.6}$$

which, upon substituting back into (C.1), gives the Mercator transformation.

Appendix D. Derivation of the mapping equation

The equation that governs the mapping from (u_1, u_2) to (ξ_1, ξ_2) can be derived from (13), i.e.,

$$g_{22}\frac{\partial^2\mathbf{r}}{\partial\xi_1^2} - 2g_{12}\frac{\partial^2\mathbf{r}}{\partial\xi_1\,\partial\xi_2} + g_{11}\frac{\partial^2\mathbf{r}}{\partial\xi_2^2} + G_3\left[\Delta_2\xi_1\frac{\partial\mathbf{r}}{\partial\xi_1} + \Delta_2\xi_2\frac{\partial\mathbf{r}}{\partial\xi_2}\right] = \mathbf{n}G_3 2H. \tag{D.1}$$

Upon applying chain rule, $\partial/\partial\xi_\alpha = \partial/\partial u_1(\partial u_1/\partial\xi_\alpha) + \partial/\partial u_2(\partial u_2/\partial\xi_\alpha)$ $(\alpha = 1, 2)$, the first three terms on the LHS leads to

$$g_{22}\frac{\partial^2\mathbf{r}}{\partial\xi_1^2} - 2g_{12}\frac{\partial^2\mathbf{r}}{\partial\xi_1\,\partial\xi_2} + g_{11}\frac{\partial^2\mathbf{r}}{\partial\xi_2^2} = J^2\left(\bar{g}_{22}\frac{\partial^2\mathbf{r}}{\partial u_1^2} - 2\bar{g}_{12}\frac{\partial^2\mathbf{r}}{\partial u_1\,\partial u_2} + \bar{g}_{11}\frac{\partial^2\mathbf{r}}{\partial u_2^2}\right) + \mathscr{L}_1 u_1\frac{\partial\mathbf{r}}{\partial u_1} + \mathscr{L}_1 u_2\frac{\partial\mathbf{r}}{\partial u_2}, \tag{D.2}$$

where the operator $\mathscr{L}_1$ stands for

$$\mathscr{L}_1 = g_{22}\frac{\partial^2}{\partial\xi_1^2} - 2g_{12}\frac{\partial^2}{\partial\xi_1\,\partial\xi_2} + g_{11}\frac{\partial^2}{\partial\xi_2^2}. \tag{D.3}$$

Similarly, the last two terms on the LHS yields

$$G_3\left\{\Delta_2\xi_1\frac{\partial\mathbf{r}}{\partial\xi_1} + \Delta_2\xi_2\frac{\partial\mathbf{r}}{\partial\xi_2}\right\} = G_3\left\{\mathscr{L}_2 u_1\frac{\partial\mathbf{r}}{\partial u_1} + \mathscr{L}_2 u_1\frac{\partial\mathbf{r}}{\partial u_2}\right\}, \tag{D.4}$$

where the operator $\mathscr{L}_2$ stands for

$$\mathscr{L}_2 = \Delta_2\xi_1\frac{\partial}{\partial\xi_1} + \Delta_2\xi_2\frac{\partial}{\partial\xi_2}. \tag{D.5}$$

Since $\mathbf{n}(k_{\mathrm{I}} + k_{\mathrm{II}})$ in (ξ_1, ξ_2) are identical to $\bar{\mathbf{n}}(\bar{k}_{\mathrm{I}} + \bar{k}_{\mathrm{II}}) = \bar{\Delta}_2\mathbf{r}$ in (u_1, u_2), i.e., they are invariant quantities, we have on the RHS

$$G_3\bar{\Delta}_2\mathbf{r} = J^2\left\{\bar{\Delta}_2 u_1\frac{\partial\mathbf{r}}{\partial u_1} + \bar{\Delta}_2 u_2\frac{\partial\mathbf{r}}{\partial u_2} + \left[\bar{g}_{22}\frac{\partial^2\mathbf{r}}{\partial u_1^2} - 2\bar{g}_{12}\frac{\partial^2\mathbf{r}}{\partial u_1\,\partial u_2} + \bar{g}_{11}\frac{\partial^2\mathbf{r}}{\partial u_2^2}\right]\right\}. \tag{D.6}$$

(Note that the relation $G_3 = \bar{G}_3 J^2$ has been used.) The functions $\bar{\Delta}_2 u_1$ and $\bar{\Delta}_2 u_2$ in (D.6) are given in (32). Upon regrouping (D.2), (D.4) and (D.6) yields

$$\left[\mathscr{L}_1 u_1 - J^2\bar{\Delta}_2 u_1 + G_3\mathscr{L}_2 u_1\right]\frac{\partial\mathbf{r}}{\partial u_1} + \left[\mathscr{L}_1 u_2 - J^2\bar{\Delta}_2 u_2 + G_3\mathscr{L}_2 u_2\right]\frac{\partial\mathbf{r}}{\partial u_2} = 0. \tag{D.7}$$

The transient term $\partial/\partial t$ is then incorporated into (D.7) by setting

$$\Delta_2\xi_1 = P + \frac{c}{G_3}\frac{\partial\xi_1}{\partial t}, \quad \Delta_2\xi_2 = Q + \frac{c}{G_3}\frac{\partial\xi_2}{\partial t},$$

$$\frac{\partial u_1}{\partial t} = -\left(\frac{\partial u_1}{\partial\xi_1}\frac{\partial\xi_1}{\partial t} + \frac{\partial u_1}{\partial\xi_2}\frac{\partial\xi_2}{\partial t}\right), \quad \frac{\partial u_2}{\partial t} = -\left(\frac{\partial u_2}{\partial\xi_1}\frac{\partial\xi_1}{\partial t} + \frac{\partial u_2}{\partial\xi_2}\frac{\partial\xi_2}{\partial t}\right), \tag{D.8}$$

and substituting into the operator $\mathscr{L}_2$ in (D.4) to give

$$\left[\mathscr{L}_1 u_1 - J^2\bar{\Delta}_2 u_1 + G_3\left(P\frac{\partial u_1}{\partial\xi_1} + Q\frac{\partial u_1}{\partial\xi_2} - \frac{c}{G_3}\frac{\partial u_1}{\partial t}\right)\right]\frac{\partial\mathbf{r}}{\partial u_1} + \left[\mathscr{L}_1 u_2 - J^2\bar{\Delta}_2 u_2 + G_3\left(P\frac{\partial u_2}{\partial\xi_1} + Q\frac{\partial u_2}{\partial\xi_2} - \frac{c}{G_3}\frac{\partial u_2}{\partial t}\right)\right]\frac{\partial\mathbf{r}}{\partial u_2} = 0. \tag{D.9}$$

Upon equating the coefficients of $\partial\mathbf{r}/\partial u_1$ and $\partial\mathbf{r}/\partial u_2$ to zero yields the pair of equations governing u_1 and u_2, respectively. Applying the orthogonal and isothermal properties into (D.9) and setting $P = Q = 0, \sigma = t/c$ yields (31).

References

[1] J.D. Anderson Jr., Computational Fluid Dynamics: The Basics with Applications, first ed., McGraw-Hill, New York, 1995.

[2] J.G. Blom, J.G. Verwer, VLUGR2: A Vectorized Local Uniform Grid Refinement Code for PDEs in 2D, Technical Report NM-R9306, CWI, Amsterdam, 1993.

[3] W.J. Chyu, S.S. Davis, K.S. Chang, Calculation of unsteady transonic flow over an airfoil, AIAA J. 19 (1981) 684.

[4] F.W. Cummings, On surface geometry coupled to morphogen, J. Theor. Biol. 137 (1989) 215.

[5] F.W. Cummings, Aspects of growth and form, Physica D 79 (1994) 146.

[6] F.W. Cummings, Geometrical concepts in epithelial sheets, J. Theor. Biol. 179 (1996) 41.

[7] F.W. Cummings, A model of growth and form based on adhesion molecules, J. Theor. Biol. 178 (1996) 229.

[8] F.W. Cummings, Spatial patterning via PTP adhesive phosphatases, J. Theor. Biol. 196 (1999) 19.

[9] F.W. Cummings, A model of pattern formation based on signalling pathways, J. Theor. Biol. 207 (2000) 107.

[10] F.W. Cummings, private communication.

[11] F.W. Cummings, The interaction of surface geometry with morphogens, J. Theor. Biol. 212 (2001) 303.

[12] R. Dillon, H.G. Othmer, A mathematical model for outgrowth and spatial patterning of the vertebrate limb bud, J. Theor. Biol. 197 (1999) 295.

[13] M.P. DoCarmo, Differential Geometry of Curves and Surfaces, first ed., Prentice-Hall, Englewood Cliffs, NJ, 1976.

[14] L. Eça, Handbook of Grid Generation, first ed., CRC Press, London, 1999 (Chapter Seven).

[15] H.J. Haussling, Boundary-fitted coordinate for accurate numerical solution of multibody flow problems, J. Comput. Phys. 30 (1979) 107.

[16] A. Khamayseh, A. Kuprat, Handbook of Grid Generation, first ed., CRC Press, London, 1999 (Chapter Nine).

[17] A. Khamayseh, A. Kuprat, C.W. Mastin, Handbook of Grid Generation, first ed., CRC Press, London, 1999 (Chapter Six).

[18] A. Khamayseh, C.W. Mastin, Computational conformal mapping for surface grid generation, J. Comput. Phys. 123 (1996) 394.

[19] P.M. Knupp, A robust elliptic grid generator, J. Comput. Phys. 100 (1992) 409.

[20] A.J. Koch, H. Meinhardt, Biological pattern formation: from basic mechanisms to complex structures, Rev. Mod. Phys. 66 (4) (1994) 1481.

[21] G.P. Koomullil, Z.U.A. Warsi, Numerical mapping of arbitrary domain using spectral method, J. Comput. Phys. 104 (1993) 251.

[22] M.M. Lipschutz, Schaum's Outline Series: Differential Geometry, first ed., McGraw-Hill, New York, 1969.

[23] P.K. Maini, K.J. Painter, H.N.P. Chau, Spatial pattern formation in chemical and biological systems, J. Chem. Soc., Faraday Trans. 93 (20) (1997) 3601.

[24] H. Meinhardt, A model for pattern formation of hypostome, tentacles, and foot in hydra: how to form structures close to each other, how to form them at a distance, Dev. Biol. 157 (1993) 321.

[25] H. Meinhardt, On Growth and Form, first ed., Wiley, New York, 1999 (Chapter Eight).

[26] J.D. Murray, A pre-pattern formation mechanism for animal coat markings, J. Theor. Biol. 88 (1981) 161.

[27] G. Ryskin, L.G. Leal, Orthogonal mapping, J. Comput. Phys. 50 (1983) 71.

[28] W.H. Schiesser, The Numerical Method of Lines: Integration of Partial Differential Equations, first ed., Academic Press, New York, 1991.

[29] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 15 (1974) 299.

[30] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundations and Applications, first ed., North-Holland, New York, 1985.

[31] A.M. Turing, The chemical basis of morphogensis, Philos. Trans. R. Soc. Lond. B 327 (1952) 37.

[32] Z.U.A. Warsi, Conservation form of the Navier–Stokes equations in general nonsteady coordinates, AIAA J. 19 (2) (1981) 240.

[33] Z.U.A. Warsi, A note on the mathematical formulation of the problem of numerical coordinate generation, Quart. Appl. Math. 41 (1983) 221.

[34] Z.U.A. Warsi, Numerical grid generation in arbitrary surfaces through a second-order differential-geometric model, J. Comput. Phys. 64 (1986) 82.

[35] Z.U.A. Warsi, Theoretical foundation of the equations for the generation of surface coordinate, AIAA J. 28 (6) (1990) 1140.

[36] Z.U.A. Warsi, Fluid Dynamics: Theoretical and Computational Approaches, second ed., CRC Press, Boca Raton, FL, 1999.

[37] Z.U.A. Warsi, Handbook of Grid Generation, first ed., CRC Press, London, 1999 (Chapter Two).